Title
Cycle flow formulation of optimal network flow problems for centralized and decentralized solvers

Authors
Asadi, Reza
Kia, Solmaz S

Publication Date
2017-06-02

Cycle flow formulation of optimal network flow problems for centralized and decentralized solvers

Reza Asadi and Solmaz S. Kia

Abstract

When the underlying physical network layer in optimal network flow problems is a large graph, the associated optimization problem has a large set of decision variables. In this paper, we discuss how the cycle basis from graph theory can be used to reduce the size of this decision variable space. The idea is to eliminate the aggregated flow conservation constraint of these problems by explicitly characterizing its solutions in terms of the span of the columns of the transpose of a fundamental cycle basis matrix of the network plus a particular solution. We show that for any given input/output flow vector, a particular solution can be efficiently constructed from tracing any path that connects a source node to a sink node. We demonstrate our results over a minimum cost flow problem as well as an optimal power flow problem with storage and generation at the nodes. We also show that the new formulation of the minimum cost flow problem based on the cycle basis variables is amenable to a distributed solution. In this regard, we apply our method over a distributed alternating direction method of multipliers (ADMM) solution and demonstrate it over a numerical example.

I. Introduction

In a network flow problem, a physical system consisted of several routes between source and sink points transfers input flows from the source points to the sink points. The objective of optimal network flow problems mainly is to minimize the overall cost of transporting flow [1]. Network flow problems appear in many important applications, such as communication networks [2], wireless sensor networks [3], wireless routing and resource allocation [4], transportation systems [5] and power networks [6]. In power network problems, variants of optimal network flow problems also include optimal generation and storage costs in their objectives [7], [8], [9], [10].

The first author is a Ph.D. candidate with the Computer Science Department, the second author is an assistant professor with the Mechanical and Aerospace Engineering Department of University of California Irvine, Irvine, CA 92697 {rasadi, solmaz}@uci.edu

With the advent of new technologies, the amount of available data and size of networks have been increasing. Such expansions in the size of physical networks result in increasing the size of optimization problems associated with optimal network flow problems. The number of decision variables has a direct relation with the time and space computation complexity of optimization problems. For large scale optimization problems, there has been efforts to use different variable reduction techniques to reduce problem size. Variable fixing techniques [11], dominance technique [12] and constraint pairing techniques [13] are some general reduction techniques in Integer Quadratic Problems (IQP). Moreover, in [14] a new variable reduction techniques for IQP proposed which fixes some decision variables at zero without loosing optimality. In multi-objective optimization problems also it is shown that using data mining techniques it is possible to reduce less effective variables [15]. For evolutionary optimization problems, [16] presents how variable reduction techniques can be applied to obtain the variable relations from the partial derivatives of an optimization function. For optimization problems of the form (1), eliminating affine equality constraint as discussed below is also a method to reduce the number of the search variables of the problem (c.f. [17])–

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}}\ \phi(\mathbf{x}), \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \tag{1}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are the cost function and the inequality constraint function, respectively, and $\mathbf{A} \in \mathbb{R}^{p\times n}$ satisfies $\operatorname{rank}(\mathbf{A}) = \rho \leq p < n$. The affine feasible set for this optimization problem can be characterized as

$$\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}\} = \{\mathbf{F}\mathbf{z} + \mathbf{x}^{\mathrm{p}} \mid \mathbf{z} \in \mathbb{R}^{n-\rho}\}. \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{n\times(n-\rho)}$ is a matrix whose columns expand the null-space of $\mathbf{A}$ and $\mathbf{x}_{\mathrm{p}} \in \mathbb{R}^n$ is a particular solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then, $\mathbf{x}^\star$ in (1) satisfies $\mathbf{x}^\star = \mathbf{F}\mathbf{z}^\star + \mathbf{x}_{\mathrm{p}}$ where

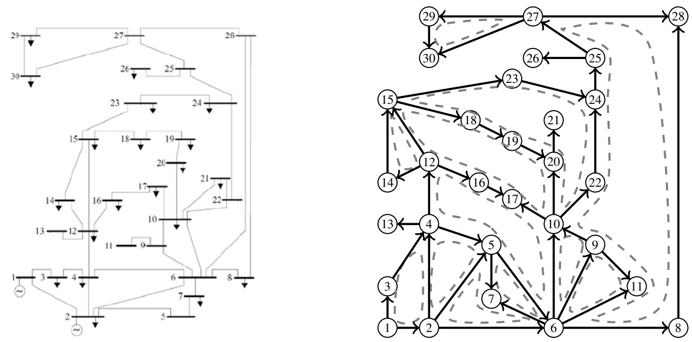
$$\begin{aligned} \mathbf{z}^\star =& \underset{\mathbf{z}\in\mathbb{R}^{n-\rho}}{\operatorname{argmin}}\ \bar{\phi}(\mathbf{z}) = \phi(\mathbf{F}\mathbf{z} + \mathbf{x}^{\mathrm{p}}), \quad \text{s.t.} \\ & \bar{\mathbf{g}}(\mathbf{z}) = \mathbf{g}(\mathbf{F}\mathbf{z} + \mathbf{x}^{\mathrm{p}}) \leq \mathbf{0}. \end{aligned} \tag{3}$$

Compared to (1), in (3) not only the equality constraint is eliminated but also the number of the search variables are reduced from n to $n-\rho$. However, the lack of efficient methods to construct matrix $\mathbf{F}$ and particular solution $\mathbf{x}_{\mathrm{p}}$ can be an impediment in use of affine equality constraint elimination method.

Optimal network flow problems are normally cast as a convex optimization problem where the cost is the sum of convex cost of flow through the arcs subject to capacity bounds for each arc and flow conservation equations at each node, resulting in optimization problems of the form (1). In variations of the optimal network flow problem, the cost can be augmented to include the cost of e.g., generation and storage at nodes. The constraints can also be expanded to include other components of the problem. Nevertheless, in all network flow problems, an affine equality constraint that is always present is the flow conservation equation. To reduce the decision variables, one can use the aforementioned affine equality elimination approach, to eliminate the aggregated flow conservation equation from the network flow problems. In this paper, we discuss how the cycle basis structure from graph theory (c.f. [18]) can be used to accomplish this elimination in an efficient manner. Minimum cycle bases have applications in many areas such as electrical circuit theory [19], structural engineering [20], surface reconstruction [21].

In this paper, we show that all the solutions of the flow conservation equation is characterized explicitly in terms of the span of the columns of the transpose of the fundamental cycles basis matrix of the network plus a particular solution. The fundamental cycle basis of a graph can be computed in polynomial time using efficient algorithms such as those in [22], [23], [24] (see Appendix for a breif review). To compute a particular solution, we show that for any given input/output flow vector, a particular solution can be efficiently constructed from a set of elementary solutions each obtained from tracing a flow of value 1 over the network from each node to a common particular sink node. We demonstrate our flow conservation equation elimination over a minimum cost flow problem as well as an optimal power flow problem with storage and generation at the nodes.

Parallel and distributed solutions are also sought as a method to solve large scale optimal network flow problems in an efficient manner. For example, a minimum cost network flow problem is solved in a distributed manner via dual sub-gradient descent in [1]. For the same problem, a distributed second order method with a better convergence rate is proposed in [25]. In [26], a distributed algorithm based on the local domain ADMM approach is proposed for minimum cost flow problem. These algorithms are all arc-based, i.e., to solve the network flow problem in a distributed manner, each arc or group of arcs are assigned to cyber-layer nodes. Then, the minimum cost network flow optimization problem is cast in a separable manner and solved by cyber-layer nodes in a cooperative way. Although in distributed algorithms the computational cost of the optimal flow problem is distributed among the cyber-layer nodes,



(a) IEEE bus system 30

(b) Node-arc representation of IEEE bus system 30 with the cycles highlighted by dashed curves

Figure 1: The graph related to IEEE bus system 30 with 41 arcs, 30 nodes and 12 cycles.

the high number of decision variables normally translates to the high number of cyber nodes or large communication overhead between neighboring cyber nodes. Our next contribution in this paper is to show that the new formulation of the minimum cost network flow problem based on the cycle basis variables, which has a reduced set of search variables, is amenable to distributed solutions. Specifically, we demonstrate implantation of a distributed ADMM solution method (c.f. [27] and [28]) over this new formulation. To implement this distributed solution we propose a cyber-layer whose nodes are defined based on the fundamental cycles of a cycle basis of the physical-layer graph. A preliminary version of parts of our results in this paper has appeared in [29].

Notations: $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{\leq 0}$ denote the set of real, positive real, non-negative real, and non-positive real numbers, respectively. We let $\mathbf{A}^\top$ be the transpose of a matrix $\mathbf{A}$ and $[\mathbf{A}]_i$ indicate its i^{th} column. We let $[\{z_k\}_{k=1}^n]$ be the column vector obtained from stacking the elements of an ordered set $\{z_k\}_{k=1}^n$. For network variables $\{p_i\}_{i=1}^N \subset \mathbb{R}$, defined over N nodes, N arcs, or N cycles, we represent the aggregate vector of these variables by $\mathbf{p} = [\{p_i\}_{i=1}^N] \in \mathbb{R}^N$.

II. A REVIEW OF CYCLE BASIS IN GRAPHS

In this section, following [18], we review our graph related terminology and conventions. We also introduce our graph related notations. We represent a graph of n nodes and m arcs with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the node set and $\mathcal{E} = \{e_1, \cdots, e_m\} \in \mathcal{V} \times \mathcal{V}$ is the arc set. The graph is assumed to be undirected and with no self-loop. A *walk* is an alternating sequence of nodes and connecting arcs. A *path* is a walk that does not include any node twice, except for its first and last nodes which can be the same. A graph is *connected* if there is a path from its every node to every other node. The degree of a node in a graph is the total number of arcs connected to that node. When there is an orientation assigned to the arcs of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we represent the *oriented graph* by $\mathcal{G}^{\mathrm{o}} = (\mathcal{V}, \mathcal{E}^{\mathrm{o}})$. We write $e_k = (v_i, v_j) \in \mathcal{E}^{\mathrm{o}}$ if arc e_k points from node v_i towards node v_j. If $(v_i, v_j) \in \mathcal{E}^{\mathrm{o}}$ then $(v_j, v_i) \notin \mathcal{E}^{\mathrm{o}}$, i.e., there is no symmetric arc in the oriented graph. For $\mathcal{G}^{\mathrm{o}}$, the *oriented incidence* matrix is the matrix $\mathfrak{I}^{\mathrm{o}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, where $\mathfrak{I}^{\mathrm{o}}_{ij} = 1$ if arc e_j leaves node v_i, $\mathfrak{I}^{\mathrm{o}}_{ij} = -1$ if arc e_j enters node v_i, otherwise $\mathfrak{I}^{\mathrm{o}}_{ij} = 0$. For a connected graph of n nodes with a given orientation, the rank of $\mathfrak{I}^{\mathrm{o}}$ is $n - 1$.

A *cycle* of $\mathcal{G}$ is any sub-graph in which each node has even degree. A *simple cycle* is a path that begins and ends on the same node with no other repetitions of nodes. A cycle vector $\mathbf{c} \in \mathbb{R}^m$ is a binary vector with $\mathbf{c}_i = 1$ if e_i is in the cycle and $\mathbf{c}_i = 0$, otherwise. A *cycle basis* of $\mathcal{G}$ is a set of simple cycles that forms a basis of the cycle space of $\mathcal{G}$. Every cycle in a given cycle basis is called a *fundamental cycle*. A fundamental cycle basis of a graph is constructed by its spanning tree, in a way that cycles formed by a combination of a path in the tree and a single arc outside of the tree. For every arc outside of the tree, there exist one cycle. Each cycle generated in this way is independent of other cycles, because it has one arc, not exist in other cycles (see Fig. 1). The dimension of cycle basis of a graph is $\mu = m - n + 1$. For cycles in an oriented graph $\mathcal{G}^{\mathrm{o}} = (\mathcal{V}, \mathcal{E}^{\mathrm{o}})$, we assign the counter clockwise direction as positive cycles orientation and define the *oriented cycle vector* $\mathbf{c}^{\mathrm{o}} \in \mathbb{R}^m$ with $\mathbf{c}^{\mathrm{o}}_i = 1$ if e_i is in the cycle and aligned with its direction, $\mathbf{c}^{\mathrm{o}}_i = -1$ if e_i is in the cycle but opposing the direction of the cycle and finally $\mathbf{c}^{\mathrm{o}}_i = 0$ if e_i is not in the cycle. Given a cycle basis, we define the *oriented fundamental cycle basis matrix* $\mathbf{B}^{\mathrm{of}} \in \mathbb{R}^{\mu\times m}$ as a matrix whose rows are each the transpose of the oriented cycle vector of the fundamental cycles of this cycle basis. This matrix satisfies $\mathrm{rank}(\mathbf{B}^{\mathrm{of}}) = \mu$.

Theorem II.1 (relationship between the oriented incidence matrix and an oriented cycle vector (c.f. [18])**).** *In an oriented graph $\mathcal{G}^o$, every oriented cycle vector $\mathbf{c}^o$ is orthogonal to every row of oriented incident matrix $\mathfrak{I}^o$, i.e., $\mathfrak{I}^o\,\mathbf{c}^o = \mathbf{0}_n$.* □

III. Decision variable reduction in network flow problems

In this section, we show how two well-known network flow problems can benefit from affine equality elimination method to reduce their search variables. We study our optimal network flow problems of interest over a network of n nodes where each node is connected to a subset of other nodes through some form of routes. For example, in a power network the route is a transmission line, while in a transportation network the route is the road connecting two conjunction nodes on the road map. The physical layer topology is described by a connected graph $\mathcal{G}_{\text{physic}} = (\mathcal{V}_{\text{physic}}, \mathcal{E}_{\text{physic}})$, where $|\mathcal{V}_{\text{physic}}| = n$ and $|\mathcal{E}_{\text{physic}}| = m$. The flow can travel in both directions in every route, however, we assume a pre-specified positive orientation for each route and based on it we describe the flow network in the physical layer by the oriented version of $\mathcal{G}_{\text{physic}}$, i.e., $\mathcal{G}^{\text{o}}_{\text{physic}} = (\mathcal{V}_{\text{physic}}, \mathcal{E}^{\text{o}}_{\text{physic}})$. This physical network transfers flow(s) from a set of source nodes to a set of sink nodes (see physical layers in Fig. 3 and Fig. 4), while respecting the conservation of the flow constraints, i.e., the total inflow into each node must be equal to the total outflow from that node. We let x_i be the flow across the arc $e_i \in \mathcal{E}^{\text{o}}_{\text{physic}} = \{e_1, e_2, \cdots, e_m\}$. Every arc $e_i \in \mathcal{E}^{\text{o}}_{\text{physic}}$ has a pre-specified capacity, i.e., $\mathsf{b}_i \leq x_i \leq \mathsf{c}_i$, for some known $\mathsf{b}_i, \mathsf{c}_i \in \mathbb{R}$. For any external flow f_i, we use the sign convention of $\mathsf{f}_i > 0$ for input flow and $\mathsf{f}_i < 0$ for output flow, and $\mathsf{f}_i = 0$ otherwise.

A. *Minimum cost flow problem*

We consider a minimum cost flow problem over $\mathcal{G}^{\text{o}}_{\text{physic}}$ with a given set of input and output flows at specific source and sink points. In this problem, there is a convex cost $\phi_i : \mathbb{R} \to \mathbb{R}$ associated with flow across each arc $e_i \in \mathcal{E}^{o}_{\text{physic}}$, and our objective is to find the network minimizer $\mathbf{x}^\star \in \mathbb{R}^m$ in the following optimization problem

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^m}{\operatorname{argmin}} \; \phi(\mathbf{x}) = \sum\nolimits_{i=1}^{m} \phi_i(x_i), \quad \text{s.t.}, \tag{4a}$$

$$\sum\nolimits_{j=1}^{m} \mathfrak{I}^{\text{o}}_{ij} \, x_j = \mathsf{f}_i, \quad i \in \{1, \ldots, n\}, \tag{4b}$$

$$\mathsf{b}_j \leq x_j \leq \mathsf{c}_j, \qquad j \in \{1, \cdots, m\}, \tag{4c}$$

where $\mathbf{f} = (\mathsf{f}_1, \cdots, \mathsf{f}_n)^\top$ is the given input/output flow vector which satisfies $\sum_{i=1}^{n} \mathsf{f}_i = 0$. Here, (4b) captures the flow conservation at nodes across the network and (4c) describes the arc capacity constraints. The number of search variables in the optimization problem (4) is equal to the number of the arcs of the network, i.e., $|\mathcal{E}^{o}_{\text{physic}}| = m$. Our result below uses Theorem II.1 to eliminate the affine equality constrains (4b) and reduce the search variables to $m - n + 1$.

Theorem III.1 (Eliminating the flow conservation constraint from (4))**.** *Consider the optimal network flow problem* (4) *over a connected physical network* $\mathcal{G}^o_{\text{physic}}$*. Then,* $\mathbf{x}^\star$ *in* (1) *satisfies* $\mathbf{x}^\star = \mathbf{B}^{\text{of}\,\top}\mathbf{z}^\star + \mathbf{x}^{\text{p}}$ *where*

$$\mathbf{z}^\star = \underset{\mathbf{z}\in\mathbb{R}^{m-n+1}}{\operatorname{argmin}}\, \phi(\mathbf{z}) = \sum_{i=1}^{m} \phi_i(\mathbf{z}^\top[\mathbf{B}^{\text{of}}]_i + \mathsf{x}^p_i), \quad \textit{s.t.} \tag{5}$$

$$\mathsf{b}_j \leq \mathbf{z}^\top[\mathbf{B}^{\text{of}}]_j + \mathsf{x}^p_j \leq \mathsf{c}_j, \quad j \in \{1, \cdots, m\},$$

and $\mathbf{B}^{\text{of}}$ *is an oriented fundamental cycle matrix of* $\mathcal{G}^o_{\textit{physic}}$ *and* $\mathbf{x}^{\text{p}}$ *is a particular solution of* $\mathfrak{I}^o\,\mathbf{x} = \mathbf{f}$.

Proof. The equality constraint (4b) in aggregated form is

$$\mathfrak{I}^{\text{o}}\,\mathbf{x} = \mathbf{f}. \tag{6}$$

Invoking the same argument that is used to relate solutions of the optimization problem (1) to those of (3), the proof relays on showing that the null-space of $\mathfrak{I}^{\text{o}}$ is spanned by columns of $\mathbf{B}^{\text{of}\,\top}$. By virtue of Theorem II.1, we have $\mathfrak{I}^{\text{o}}\mathbf{B}^{\text{of}\,\top} = \mathbf{0}$. Recall that for a connected oriented graph $\operatorname{rank}(\mathfrak{I}^{\text{o}}) = n-1$. Because $\operatorname{rank}(\mathbf{B}^{\text{of}\,\top}) = m-n+1$, null-space of $\mathfrak{I}^{\text{o}} \in \mathbb{R}^{n\times m}$ is spanned by columns of $\mathbf{B}^{\text{of}\,\top}$. This completes our proof. □

The effectiveness of the decision variable reduction method in Theorem III.1 depends on how efficiently one can construct matrix $\mathbf{B}^{\text{of}}$ and particular solution $\mathbf{x}^{\text{p}}$, especially in large scale networks. In regards to matrix $\mathbf{B}^{\text{of}}$, as reviewed in Appendix, there are efficient algorithms that can construct cycle basis in polynomial time. Next, we propose a simple method to construct a particular solution $\mathbf{x}^{\text{p}}$ using graph topology. Our method relies on the superposition property of linear algebra equations, and the fact that a particular solution for a unit flow $\mathsf{f}_i = 1$ entering the network at node v_i and leaving it at node v_j can simply be constructed by assuming that $\mathsf{f}_i = 1$ flows along a path from node v_i to node v_j.

Lemma III.1 (Particular solution of (6))**.** *Given an input/output flow vector* $\mathbf{f}$ *over* $\mathcal{G}^o_{\textit{physic}}$ *which satisfies* $\sum_{i=1}^{n} \mathsf{f}_i = 0$*, a particular solution for* (6) *is* $\mathbf{x}^p = \sum_{i=1}^{n-1} \mathsf{f}_i\, \bar{\mathbf{x}}^{p,v_i}$*. Here,* $\bar{\mathbf{x}}^{p,v_i} \in \mathbb{R}^m$*,* $i \in \{1, \dots, n-1\}$*, is constructed from a path that connects node* v_i *to node* v_n *such that* $\mathsf{x}^{p,v_i}_j = 1$ *(resp.* $\mathsf{x}^{p,v_i}_j = -1$*) if* e_j*,* $j \in \{1, \dots, m\}$*, is on this path and is along (resp. opposing) the direction of the path, otherwise* $\mathsf{x}^{p,v_i}_j = 0$.

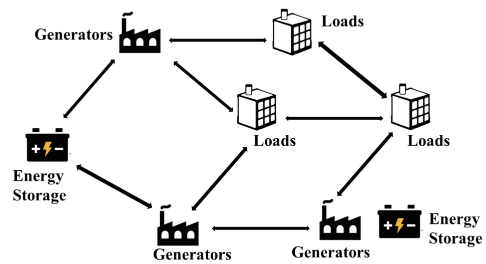


Figure 2: An schematic representation of a network flow problem with generation and storage at nodes

Proof. For every $i \in \{1,\dots,n-1\}$, consider a virtual scenario where a unit flow $\bar{\mathsf{f}}_i = 1$ enters the network at node v_i and leaves it at node v_n. Using a simple flow tracing over the network we can see that $\bar{\mathbf{x}}^{\mathrm{p},v_i} \in \mathbb{R}^m$ as described in the statement satisfies $\mathfrak{I}^{\mathrm{o}}\,\bar{\mathbf{x}}^{\mathrm{p},v_i} = \bar{\mathbf{f}}^{v_i}$, $i \in \{1,\dots,n-1\}$ where $\bar{\mathsf{f}}_i^{v_i} = 1$, $\bar{\mathsf{f}}_n^{v_i} = -1$ and $\bar{\mathsf{f}}_j^{v_i} = 0$, $j \in \{1,\dots,n\}\backslash\{i,n\}$. For a given network flow vector $\mathbf{f}$ because $\mathsf{f}_n = -\sum_{i=1}^{n-1}\mathsf{f}_i$, we can write $\mathbf{f} = \sum_{i=1}^{n-1}\mathsf{f}_i\,\bar{\mathbf{f}}^{v_i}$. Therefore, $\mathfrak{I}^{\mathrm{o}}\sum_{i=1}^{n-1}\mathsf{f}_i\,\bar{\mathbf{x}}^{\mathrm{p}}_{v_i} = \sum_{i=1}^{n-1}\mathsf{f}_i\,\bar{\mathbf{f}}^{v_i} = \mathbf{f}$, which completes our proof. $\square$

A few remarks are in order regarding the particular solution. First, note that construction of the 'elementary' particular solution set $\{\bar{\mathbf{x}}^{\mathrm{p},v_i}\}_{i=1}^{n-1}$ is regardless of the value of the network flow vector $\mathbf{f}$. Second, for problems with single source and single sink nodes, we can label the source node v_1 and the sink node v_n and compute the elementary solution set only for node v_1. More particularly, if in a given network flow problem the sink and the source nodes are fixed we only need to compute the elementary particulars solution set for the collection of sink and source nodes (see Section IV-B for an illustrative numerical example). Finally, to obtain sparse elementary solutions, we can use a shortest path between nodes v_i and v_n.

B. Optimal power flow with storage and generation at nodes

Next, we consider an optimal power flow problem over a network described by $\mathcal{G}^{\mathrm{o}}_{\text{physic}}$ with storage, generation and load at its nodes (see Fig. 2). The objective in this problem is to minimize the cost of power generation along with energy loss at the transmission lines over some finite time interval $\mathcal{T} = \{1,\cdots,T\}$. Mathematical modeling of this problem over various scenarios including deterministic and stochastic generators is considered in the literature, e.g., [7], [8], [9], [10]. All these models, at each time $t \in \mathcal{T}$, include a flow conservation equation at each node.

As a result for a network with m arcs, the flow conservation equation introduces $m\,|\mathcal{T}|$ decision variables into the optimization problem.

In our study below, without loss of generality, we use the deterministic form (no renewable generation source) of the optimal network flow problem studied in [7], which states the problem as a direct current (DC) power flow problem (see (7)). Without loss of generality, we assume that at each time $t \in \mathcal{T}$, each node $v_i \in \mathcal{V}_{\text{physic}}$ has a generator which supplies a bounded $\delta_i(t)$ power, a battery with a bounded storage level $s_i(t)$ and a charge/discharge variable $u_i(t)$, and a known demand $\mathsf{d}_i(t) \in \mathbb{R}_{\leq 0}$. If a node does not have either of the generation, storage, or load components, we simply remove the respective variables from our formulation below. Then, given a known load profile $\{\mathbf{d}(t)\}_{t=1}^{T}$, where $\mathbf{d}(t) = [\{\mathsf{d}_i(t)\}_{i=1}^{n}]$, the optimization problem of interest is

$$\{\mathbf{x}^{\star}(t), \boldsymbol{\delta}^{\star}(t), \mathbf{u}^{\star}(t), \mathbf{s}^{\star}(t), \boldsymbol{\theta}^{\star}(t)\}_{t=1}^{T} = \tag{7a}$$

$$\text{argmin}\ \frac{1}{T}\sum\nolimits_{t=1}^{T}\Big(\sum\nolimits_{j=1}^{n} g_j(\delta_j(t)) + \sum\nolimits_{i=1}^{m} \phi_i(x_i(t))\Big), \text{ s.t.}$$

for $t \in \mathcal{T}, \quad i \in \{1, \ldots, n\}$

$$\sum\nolimits_{j=1}^{m} \mathfrak{I}^{\text{o}}_{ij}\, x_i(t) = \delta_i(t) + u_i(t) + \mathsf{d}_i(t), \tag{7b}$$

$$s_i(t+1) = \lambda_i s_i(t) + u_i(t), \tag{7c}$$

$$\mathbf{B}_{ij}(\theta_i(t) - \theta_j(t)) = x_i(t), \quad j \in \mathcal{N}^e(i), \tag{7d}$$

$$\underline{\delta}_i \leq \delta_i(t) \leq \bar{\delta}_i, \quad \underline{\mathsf{u}}_i \leq u_i(t) \leq \bar{\mathsf{u}}_i, \quad \underline{\mathsf{s}}_i \leq s_i(t) \leq \bar{\mathsf{s}}_i, \tag{7e}$$

$$\mathsf{b}_j \leq x_j(t) \leq \mathsf{c}_j, \qquad j \in \{1, \cdots, m\}, \tag{7f}$$

where, g_j, $j \in \{1, \ldots, n\}$ and ϕ_i, $i \in \{1, \ldots, m\}$ are cost functions for generators and power flows, respectively. Here, $\mathcal{N}^e(i)$ is the set of the nodes that are connected to node v_i through an arc, and $\lambda_i \in (0, 1]$ is the storage energy dissipation factor. Moreover, $(\underline{\mathsf{s}}_i, \bar{\mathsf{s}}_i) \in \mathbb{R} \times \mathbb{R}$, $(\underline{\mathsf{u}}_i, \bar{\mathsf{u}}_i) \in \mathbb{R} \times \mathbb{R}$, $(\underline{\delta}_i, \bar{\delta}_i) \in \mathbb{R} \times \mathbb{R}$, are, respectively, known (lower bound, upper bound) values on storage level, battery charge/discharge and power generation by the generator. Finally $\mathbf{B}_{ij}(\theta_i(t) - \theta_j(t))$ is the DC approximation for alternating current power flow. Here, $\theta_i(t)$ is the voltage phase angle of node (bus) $v_i \in \mathcal{V}_{\text{physic}}$ at time t and $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the imaginary part of the admittance matrix under DC assumption.

The following result shows that, the number of search variables related to the flow across the arcs in the optimization problem (7) can be reduced from $m|\mathcal{T}|$ to $(m-n+1)\mathcal{T}$ via eliminating the flow conservation constraint at each $t \in \mathcal{T}$. An interesting, observation in the result below is that in order to eliminate the flow conservation equations (7b), we need to introduce a new set of affine equality constraint (8b) which ensure balance between the external input and output flows.

Proposition III.1 (Eliminating the flow conservation constraint from (7))**.** *Consider the optimal power flow problem* (7) *over a physical network described by* $\mathcal{G}^o_{\text{physic}}$ *with a given set of loads* $\{\mathsf{d}(t)\}_{t=1}^T$*. Then,* $\{\mathsf{B}^{\text{of}\,\top}\mathbf{z}^\star(t) + \mathsf{x}^{\text{p}}(\boldsymbol{\delta}^\star(t), \mathbf{u}^\star(t), \mathsf{d}(t)), \mathbf{u}^\star(t), \mathrm{s}^\star(t), \boldsymbol{\theta}^\star(t)\}_{t=1}^T$ *is a minimizer of the optimization problem* (7) *where*

$$\begin{aligned}
&\{\mathbf{z}^\star(t), \boldsymbol{\delta}^\star(t), \mathbf{u}^\star(t), \mathrm{s}^\star(t), \boldsymbol{\theta}^\star(t)\}_{t=1}^T = \\
&\quad \text{argmin}\ \frac{1}{T}\sum\nolimits_{t=1}^{T}\Big(\sum\nolimits_{j=1}^{n} g_j(\delta_j(t)) + \qquad\qquad \text{(8a)}\\
&\qquad\qquad \sum\nolimits_{i=1}^{m}\phi_i(\mathbf{z}(t)^\top[\mathsf{B}^{\text{of}}]_i + \mathsf{x}^p_i(\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t))\Big),\ \textit{s.t.}
\end{aligned}$$

for $t \in \mathcal{T}, \quad i \in \{1, \dots, n\}$

$$\sum\nolimits_{j=1}^{n}(\delta_j(t) + u_j(t) + \mathsf{d}_j(t)) = 0, \tag{8b}$$

$$s_i(t+1) = \lambda_i s_i(t) + u_i(t), \tag{8c}$$

$$\mathbf{B}_{ij}(\theta_i(t) - \theta_j(t)) = \mathbf{z}(t)^\top[\mathsf{B}^{\text{of}}]_i + \mathsf{x}^p_i(\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t)), \quad j \in \mathcal{N}^e(i), \tag{8d}$$

$$\underline{\delta}_i \leq \delta_i(t) \leq \bar{\delta}_i, \quad \underline{\mathsf{u}}_i \leq u_i(t) \leq \bar{\mathsf{u}}_i, \quad \underline{\mathsf{s}}_i \leq s_i(t) \leq \bar{\mathsf{s}}_i, \tag{8e}$$

$$\mathsf{b}_j \leq \mathbf{z}(t)^\top[\mathsf{B}^{\text{of}}]_j + \mathsf{x}^p_j(\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t)) \leq \mathsf{c}_j, \quad j \in \{1, \cdots, m\}. \tag{8f}$$

Here, B^{of} *is a fundamental cycle basis matrix of* $\mathcal{G}^o_{physic}$ *and* $\mathsf{x}^{\text{p}}(\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t)) = \sum_{i=1}^{n-1}(\delta_i(t) + u_i(t) + \mathsf{d}_i(t))\,\bar{\mathsf{x}}^{p,v_i}$*, where* $\{\bar{\mathsf{x}}^{p,v_i}\}_{i=1}^{n-1}$ *is as described in Lemma III.1.*

Proof. The equality constraint (7b) in aggregated form is

$$\mathfrak{I}^{\text{o}}\,\mathrm{x}(t) = \boldsymbol{\delta}(t) + \mathbf{u}(t) + \mathsf{d}(t), \qquad t \in \mathcal{T}. \tag{9}$$

Note that rank of $\mathfrak{I}^{\text{o}} \in \mathbb{R}^{n\times m}$ is $n-1$ and also that $\mathbf{1}_n^\top\mathfrak{I}^{\text{o}} = \mathbf{0}$, where $\mathbf{1}_n$ is the vector of n ones. Left multiplying (9) by $\mathbf{1}_n^\top$ results in (8b). Then, for $\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t) \in \mathbb{R}^n$ that satisfy (8b), following the method discussed in Lemma III.1, we can show that $\mathsf{x}^{\text{p}}(\boldsymbol{\delta}(t), \mathbf{u}(t), \mathsf{d}(t)) =$

$\sum_{i=1}^{n-1}(\delta_i(t)+u_i(t)+\mathsf{d}_i(t))\,\bar{\mathsf{x}}^{\mathrm{p},v_i}$ is a particular solution of (9), i.e., $\mathfrak{I}^{\mathrm{o}}\,\mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t)) = \boldsymbol{\delta}(t)+\mathbf{u}(t)+\mathbf{d}(t)$. Therefore, for any given load vector $\mathbf{d}$ (to simplify the notation we drop argument t), we have

$$\big\{\mathbf{x}\in\mathbb{R}^m \mid \mathfrak{I}^{\mathrm{o}}\mathbf{x}=\boldsymbol{\delta}+\mathbf{u}+\mathbf{d}, \boldsymbol{\delta}\in\mathbb{R}^n, \mathbf{u}\in\mathbb{R}^n\big\} =$$
$$\Big\{\mathbf{B}^{\mathrm{of}\,\top}\mathbf{z}+\mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta},\mathbf{u},\mathbf{d}) \,\Big|\, \mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta},\mathbf{u},\mathbf{d}) = \sum\nolimits_{i=1}^{n-1}(\delta_i+u_i+\mathsf{d}_i)\,\bar{\mathsf{x}}^{\mathrm{p},v_i},$$
$$\sum\nolimits_{i=1}^{n}(\delta_i+u_i+\mathsf{d}_i)=0,\, \mathbf{z}\in\mathbb{R}^{n-m+1}, \boldsymbol{\delta}\in\mathbb{R}^n, \mathbf{u}\in\mathbb{R}^n\Big\}.$$

As a result, at each $t\in\mathcal{T}$, we can eliminate the affine equation (7b) and arrive at the equivalent optimization problem (8) whose minimizers are related to minimizers of (7) in a way that is described in the statement. □

IV. A CYCLE-BASIS DISTRIBUTED ADMM ALGORITHM FOR MINIMUM COST NETWORK FLOW PROBLEM

In this section, we consider the minimum cost network flow problem (4) and show that its equivalent form (5) based on the cycle basis variables is amenable to a distributed solution. We start by introducing some notation related to the oriented fundamental cycles of $\mathcal{G}^{\mathrm{o}}$. Let $\mathcal{C}^{\mathrm{of}}=\{\mathfrak{C}_i^{\mathrm{of}}\}_{i=1}^{\mu}$, where $\mu=m-n+1$, be the set of fundamental cycles of $\mathcal{G}^{\mathrm{o}}$ whose cycle matrix $\mathbf{B}^{\mathrm{of}}$ is used to eliminate the flow conservation equation as explained in Section III. We represent the set of arcs of any $\mathfrak{C}_i^{\mathrm{of}}\in\mathcal{C}^{\mathrm{of}}$, $i\in\{1,\dots,\mu\}$, by $\mathcal{E}_i^{\mathfrak{C}}=\{e_j\in\mathcal{E}^{\mathrm{o}}, j\in\{1,\dots,m\}\,|\,\mathbf{B}_{ij}^{\mathrm{of}}\neq 0\}$. For a given cycle basis $\mathcal{C}^{\mathrm{of}}$, we refer to the cycles that share an arc as neighbors and represent the set of (cycle) neighbors of any fundamental cycle $\mathfrak{C}_i^{\mathrm{of}}\in\mathcal{C}^{\mathrm{of}}$, $i\in\{1,\dots,\mu\}$, by $\mathcal{N}_i^{\mathfrak{C}} = \{j\in\{1,\dots,\mu\}\backslash\{i\}\,|\,\exists\, k\in\{1,\dots,m\}$ s.t. $\mathbf{B}_{ik}^{\mathrm{of}}\neq 0$ and $\mathbf{B}_{jk}^{\mathrm{of}}\neq 0\}$. We let $\mathcal{C}^{\mathrm{of}}(e_i)$ be the set of indexes of the fundamental cycles that arc $e_i\in\mathcal{E}^{\mathrm{o}}$ belongs to them, i.e., $\mathcal{C}^{\mathrm{of}}(e_i)=\{j\in\{1,\dots,\mu\}\,|\,e_i\in\mathcal{E}_j^{\mathfrak{C}}\}$.

Recall that to eliminate the flow conservation equation we used $\mathbf{x}=\mathbf{B}^{\mathrm{of}\,\top}\mathbf{z}+\mathbf{x}^{\mathrm{p}}$, or equivalently $x_i=\mathbf{z}^\top[\mathbf{B}^{\mathrm{of}}]_i+\mathsf{x}_i^{\mathrm{p}}$, $i\in\{1,\dots,m\}$. Notice that one can think of every z_i, $i\in\{1,\dots,\mu\}$ as a cycle flow variable (with positive direction in counterclockwise direction) of the fundamental cycle $\mathfrak{C}_i^{\mathrm{of}}$. Recall that, for a given arc $e_i\in\mathcal{E}^{\mathrm{o}}_{\text{physic}}$, every element of $\mathbf{B}_{ji}^{\mathrm{of}}$ is zero except if cycle $\mathfrak{C}_j^{\mathrm{of}}$ contains arc e_i, i.e., $e_i\in\mathcal{E}_j^{\mathfrak{C}}$. As a result, we can deduce that every x_i, $i\in\{1,\dots,m\}$, is an affine function of $\mathsf{x}_i^{\mathrm{p}}$ and $\{z_k\}_{k\in\mathcal{C}^{\mathrm{of}}(e_i)}$, indicating that every arc flow is a function of its particular solution and the cycle flow of fundamental cycles that contain the arc. Given such relationship, then the cost function of every arc as $\phi_i(x_i)=\phi_i(\mathbf{z}^\top[\mathbf{B}^{\mathrm{of}}]_i+\mathsf{x}_i^{\mathrm{p}})=\psi_i(\{z_k\}_{k\in\mathcal{C}^{\mathrm{of}}(e_i)})$.

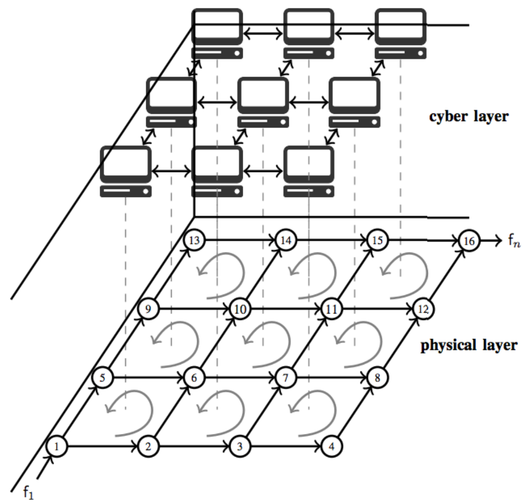


Figure 3: Physical and cyber layers of an optimal network flow problem: Physical layer has $n = 16$ nodes and $m = 24$. In this layer, the arrows indicate the positive flow directions. The cyber layer is constructed using the minimum wight cycle basis of the physical layer graph. The cyber layer has $N = 9$ agents with processing and communication capabilities.

Cyber layer architecture: based on the observation above, we propose to assign a cyber-layer node to each fundamental cycle (see Fig. 3 as an example). We assume that the cyber-layer nodes of neighboring fundamental cycles can communicate with each other in bi-directional way. For bi-connected physical layer graphs this procedure will result in a connected graph of μ nodes for cyber layer (see Fig. 3 and Fig. 4 for examples). To obtain fundamental cycles with fewest number of arcs we propose to use minimum weight cycle basis algorithms to generate the fundamental cycle basis for the cyber layer (see Appendix).

A. Cycle Basis distributed ADMM solution for minimum cost network flow problem

In this section, we derive an equivalent representation of optimization algorithm (5) which can be solved in a distributed manner using the ADMM algorithm of [26] by our cycle-based cyber layer. To this end, for every cyber-layer node $i \in \{1, \dots, \mu\}$, we define $\mathbf{y}_i = (\bar{y}_i, \tilde{\mathbf{y}}_i) \in \mathbb{R}^{|\mathcal{N}_i^{\mathfrak{C}}|+1}$, where $\bar{y}_i \in \mathbb{R}$ is the local copy of z_i and $\tilde{\mathbf{y}}_i$ is the local copy of $\{z_k\}_{k \in \mathcal{N}_i^{\mathfrak{C}}}$ at cyber node i. With this definition, we assume that every cyber node, besides its own corresponding cycle flow, has also a copy of cycle flow variable of its neighbors. Next, we cast the cost function of each cyber node in terms of its decision variable $\mathbf{y}_i$. Let $\mathbf{y}_i(e_k)$ be the component(s) of $\mathbf{y}_i$ corresponding to $\{z_j\}_{\{j\in\mathcal{C}^{\rm of}(e_k)\}}$. For every cyber-layer node, we define its cost function as

$$\theta_i(\mathbf{y}_i) \;=\; \sum_{\forall e_k\in\mathcal{E}_i^{\mathfrak{c}}} \frac{1}{|\mathcal{C}^{\rm of}(e_i)|}\psi_k(\mathbf{y}_i(e_k)). \tag{10}$$

Then, we can cast the minimum cost network flow problem (5) in the following equivalent form

$$\mathbf{y}^\star = \underset{\mathbf{y}_1,\cdots,\mathbf{y}_\mu}{\operatorname{argmin}} \sum\nolimits_{i=1}^{\mu} \theta_i(\mathbf{y}_i), \qquad \text{s.t.} \tag{11}$$

set of constraints at each cyber agent $\;i\in\{1,\dots,\mu\}$:

$$\begin{cases}\mathbf{y}_i(e_k)=[\{\bar{y}_j\}_{j\in\mathcal{C}^{\rm of}(e_i)}], \\ \mathsf{b}_k\leq \mathbf{y}_i(e_k)^\top[\{\mathbf{B}^{\rm of}_{jk}\}_{j\in\mathcal{C}^{\rm of}(e_k)}]+\mathsf{x}^{\rm p}_k\leq \mathsf{c}_k,\end{cases} \qquad \forall e_k\in\mathcal{E}_i^{\mathfrak{c}}.$$

In this formulation, every cycle-based cyber node has a copy of the cycle flows that go through its arcs, i.e, $\mathbf{y}_i$. The equality constraint at each node ensures that local copies of the cycle flows of the neighboring agents are the same, while the inequality constraint ensures that the flow through the arcs' of each cycle respect the capacity bounds.

The new formulation (11) fits the standard framework developed for distributed ADMM solutions and can be solved for example using the algorithm of [26]. The details are omitted for brevity. In this distributed implementation we assume that elementary particular solution set $\{\bar{\mathbf{x}}^{\mathrm{p},v_i}\}_{i=1}^{n-1}$ are computed off-line and are available at cyber nodes. At operation times, we only need to broadcast the input/output flow vector $\mathbf{f}$ to the cyber-layer agents. For networks with large fundamental cycle sizes, one can split a cycle among several cyber nodes. In this case the length of the $\bar{y}_i$ of these agents will be 0 and we can still use the distributed ADMM algorithm to solve the problem. Similarly, two or more cycles can be assigned to one cyber layer.

B. Numerical Example

In this section, we demonstrate the use of distributed cycle-based ADMM algorithm for a minimum cost optimal flow over the network shown in Fig. 1. We assign positive flow orientation to the arcs as represented in Fig. 4. We generate the cyber layer based on the minimum weight cycle basis as shown in the bolder network with gray nodes in Fig. 1. In this problem, we set $\mathsf{b}_i=-\mathsf{c}_i$, and $\mathsf{c}_\mathsf{i}\in\mathbb{R}_{>0}$, $i\in\{1,\dots,18\}$. We assume that the cost of the network flow at each arc is given as $\phi_i(x_i)=(\frac{x_i}{\mathsf{c}_i})^2$, where $x_i=\mathbf{z}^\top[\mathbf{B}^{\rm of}]_i+\mathsf{x}^{\rm p}_i$ is the arc flow and $\mathbf{z}=(z_1,\cdots,z_8)^\top$ are cycle flows. In the physical layer network in Fig. 4, there are two source nodes v_1 and v_4 and two two sink nodes v_9 and v_{11}.

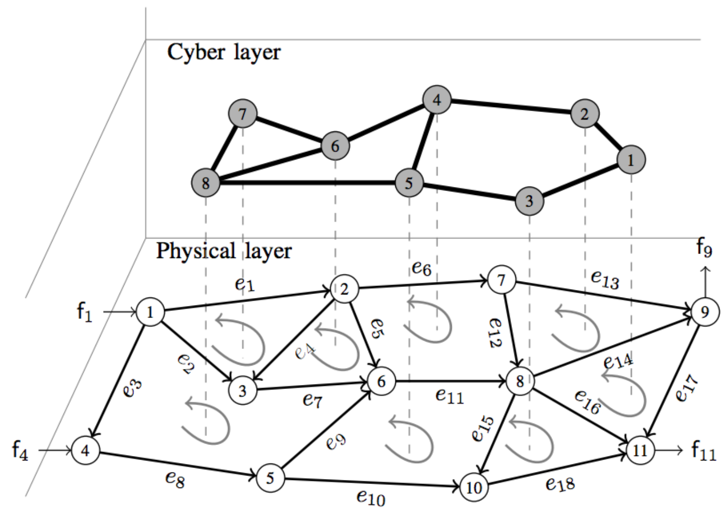
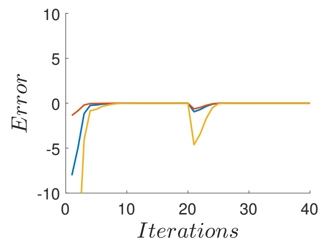


Figure 4: A physical network with a cycle-based cyber layer network overlaid atop . The physical layer network has two source nodes v_1 and v_4 and two sink nodes v_9 and v_{11}.

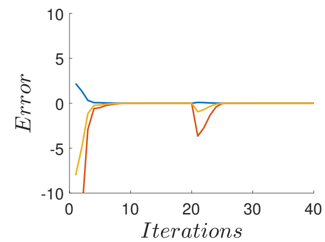
We follow Lemma III.1 to generate a particular solution for given input output flow vector $\mathbf{f} = (\mathsf{f}_1, 0, 0, \mathsf{f}_4, 0, 0, 0, 0, \mathsf{f}_9, 0, -(\mathsf{f}_1 + \mathsf{f}_4 + \mathsf{f}_9))^\top$– recall that $\mathsf{f}_{11} = -(\mathsf{f}_1 + \mathsf{f}_4 + \mathsf{f}_9)$ (recall that input flows have positive and output flows have negative values). We compute the elementary particular solutions for nodes v_1, v_4 and v_9 using shortest path from them to node v_{11}: $\bar{\mathbf{x}}^{\mathrm{p},v_1} = (0, 0, 1, \mathbf{0}_{1\times4}, 1, 0, 1, \mathbf{0}_{1\times7}, 1)^\top$, $\bar{\mathbf{x}}^{\mathrm{p},v_4} = (\mathbf{0}_{1\times7}, 1, 0, 1, \mathbf{0}_{1\times7}, 1)^\top$, and $\bar{\mathbf{x}}^{\mathrm{p},v_9} = (\mathbf{0}_{1\times16}, 1, 0)^\top$. Then $\bar{\mathbf{x}}^{\mathrm{p}} = \mathsf{f}_1\, \bar{\mathbf{x}}^{\mathrm{p},v_1} + \mathsf{f}_4\, \bar{\mathbf{x}}^{\mathrm{p},v_4} + \mathsf{f}_9\, \bar{\mathbf{x}}^{\mathrm{p}}$.

In our simulation, the lower and upper capacity bounds are selected uniformly randomly from $[2, 50]$, i.e., $\mathsf{c}_i \in [2, 50]$, $i \in \{1, \dots, m\}$. Specifically, at arcs connected to sink and source points we have $\mathsf{c}_1 = 30$, $\mathsf{c}_2 = 2$, $\mathsf{c}_3 = 27$, $\mathsf{c}_8 = 50$, $\mathsf{c}_{13} = 49$, $\mathsf{c}_{14} = 23$, $\mathsf{c}_{15} = 21$, $\mathsf{c}_{16} = 11$, $\mathsf{c}_{17} = 16$, and $\mathsf{c}_{18} = 37$. For our selected capacity bounds, using Edmonds-Krap algorithm [30] we obtain the maximum input flow $\mathsf{f}_1 + \mathsf{f}_4$ to be 82. In our simulation, then we set $\mathsf{f}_1 = 32$ and $\mathsf{f}_4 = 50$. The output follows are $\mathsf{f}_9 = -52$ and $\mathsf{f}_{11} = -30$. After, 50 iteration the input/output flows change to $\mathsf{f}_1 = 15$, $\mathsf{f}_4 = 30$, $\mathsf{f}_9 = -15$, and $\mathsf{f}_{11} = -30$.

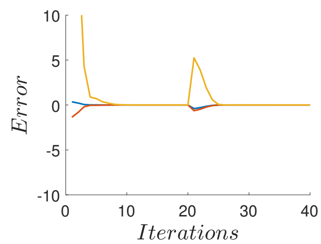
We use Matlab 'quadprog' to solve the problem in a centralized manner to generate reference values to compare the performance of our distributed cycle basis distributed ADMM algorithm as outlined in Section IV. The results are depicted in Fig. 5. In Fig. 5, plots show the distance of arc flow values from their optimum solution during execution of distributed ADMM. During the first 50 iteration the distributed ADMM converges to the optimum solution. Then, for the



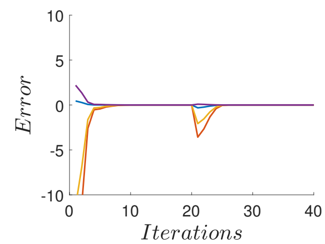
(a) cyber-layer node 1 (fundamental cycle 1)



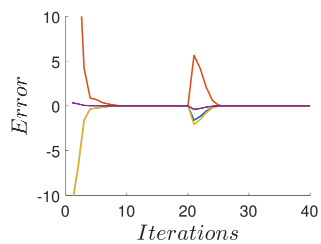
(b) cyber-layer node 2 (fundamental cycle 2)



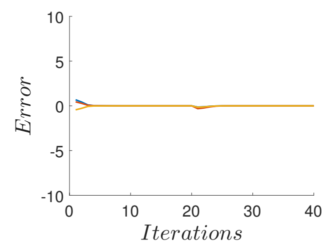
(c) cyber-layer node 3 (fundamental cycle 3)



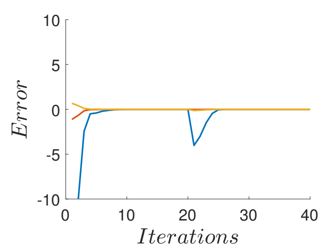
(d) cyber-layer node 4 (fundamental cycle 4)



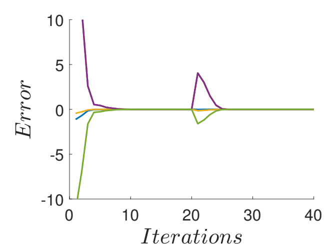
(e) cyber-layer node 5 (fundamental cycle 5)



(f) cyber-layer node 6 (fundamental cycle 6)



(g) cyber-layer node 7 (fundamental cycle 7)



(h) cyber-layer node 8 (fundamental cycle 8)

Figure 5: Each plot depicts $x_i(k) - x_i^\star$, for e_i's in that sub-captioned fundamental cycle. As this figure shows, every cyber node asymptotically calculates the optimal arc flow for its arcs.

second external flows, it converges to the optimum solution again.

V. Conclusion and future works

We considered optimal network flow problems and investigated how the decision variables of these problems can be reduced by eliminating the affine flow conservation equations. Our study was based on exploiting cycle basis concept from graph theory to eliminate flow conservation equation in an efficient manner. In particular, we showed that the computation regarding the proposed variable reduction can be done in a systematic manner, in polynomial time, using existing algorithms. Moreover, we showed that the new formulation of the optimal network flow problems with reduced variables is amenable to distributed solvers. In this regard, we constructed a cyber-layer structure based on cycles in the physical-layer network. We also demonstrated the use of a distributed ADMM solver for minimum cost flow problem.

References

[1] D. P. Bertsekas, *Network optimization: continuous and discrete models*. Citeseer, 1998.

[2] D. A. Thomas and J. F. Weng, "Minimum cost flow-dependent communication networks," *Networks*, vol. 48, no. 1, pp. 39–46, 2006.

[3] Y. Wu, A. Ribeiro, and G. B. Giannakis, "Robust routing in wireless multi-hop networks," in *Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on*, pp. 637–642, IEEE, 2007.

[4] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *Communications, IEEE Transactions on*, vol. 52, no. 7, pp. 1136–1144, 2004.

[5] Q. Ba, K. Savla, and G. Como, "Distributed optimal equilibrium selection for traffic flow over networks," in *IEEE Conference on Decision and Control*, 2015.

[6] K. Nakayama, C. Zhao, L. F. Bic, M. B. Dillencourt, and J. Brouwer, "Distributed power flow loss minimization control for future grid," *International Journal of Circuit Theory and Applications*, vol. 43, no. 9, pp. 1209–1225, 2015.

[7] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Distributed online modified greedy algorithm for networked storage operation under uncertainty," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1106–1118, 2016.

[8] K. M. Chandy, S. H. Low, U. Topcu, and H. Xu, "A simple optimal power flow model with energy storage," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 1051–1057, IEEE, 2010.

[9] S. Sun, J. A. Taylor, M. Dong, and B. Liang, "Distributed real-time phase balancing for power grids with energy storage," in *American Control Conference (ACC), 2015*, pp. 3032–3037, IEEE, 2015.

[10] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, 2012.

[11] A. Billionnet and É. Soutif, "An exact method based on lagrangian decomposition for the 0–1 quadratic knapsack problem," *European Journal of operational research*, vol. 157, no. 3, pp. 565–575, 2004.

[12] H. Kellerer, U. Pferschy, and D. Pisinger, "Other knapsack problems," in *Knapsack Problems*, pp. 389–424, Springer, 2004.

[13] M. A. Osorio, F. Glover, and P. Hammer, "Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions," *Annals of Operations Research*, vol. 117, no. 1-4, pp. 71–93, 2002.

[14] Z. Hua, B. Zhang, and X. Xu, "A new variable reduction technique for convex integer quadratic programs," *Applied Mathematical Modelling*, vol. 32, no. 2, pp. 224–231, 2008.

[15] M. Esmaeili and A. Mosavi, "Notice of retraction variable reduction for multi-objective optimization using data mining techniques; application to aerospace structures," in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, vol. 5, pp. V5–333, IEEE, 2010.

[16] G. Wu, W. Pedrycz, H. Li, D. Qiu, M. Ma, and J. Liu, "Complexity reduction in the use of evolutionary algorithms to function optimization: a variable reduction strategy," *The Scientific World Journal*, vol. 2013, 2013.

[17] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[18] A. Dharwadker and S. Pirzada, *Graph Theory*. CreateSpace Independent Publishing Platform, 2011.

[19] L. Chua and L.-K. Chen, "On optimally sparse cycle and coboundary basis for a linear graph," *IEEE Transactions on Circuit Theory*, vol. 20, no. 5, pp. 495–503, 1973.

[20] A. Cassell, J. De Henderson, and K. Ramachandran, "Cycle bases of minimal measure for the structural analysis of skeletal structures by the flexibility method," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 350, pp. 61–70, The Royal Society, 1976.

[21] G. Tewari, C. Gotsman, and S. J. Gortler, "Meshing genus-1 point clouds using discrete one-forms," *Computers & Graphics*, vol. 30, no. 6, pp. 917–926, 2006.

[22] J. D. Horton, "A polynomial-time algorithm to find the shortest cycle basis of a graph," *SIAM Journal on Computing*, vol. 16, no. 2, pp. 358–366, 1987.

[23] E. Amaldi, C. Iuliano, and R. Rizzi, "Efficient deterministic algorithms for finding a minimum cycle basis in undirected graphs," in *International Conference on Integer Programming and Combinatorial Optimization*, pp. 397–410, Springer, 2010.

[24] G. Borradaile, P. Sankowski, and C. Wulff-Nilsen, "Min st-cut oracle for planar graphs with near-linear preprocessing time," *ACM Transactions on Algorithms (TALG)*, vol. 11, no. 3, p. 16, 2015.

[25] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network flow optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 905–920, 2014.

[26] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "Distributed optimization with local domains: Applications in mpc and network flows," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, 2015.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[28] J. F. Mota, "Communication-efficient algorithms for distributed optimization," *arXiv preprint arXiv:1312.0263*, 2013.

[29] R. Asadi, S. S. Kia, and A. Regan, "Cycle basis distributed ADMM solution for optimal network flow problem over bi-connected graphs," in *54th Annual Allerton Conference on Communication, Control, and Computing*, pp. 717–723, IEEE, 2016.

[30] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.

Appendix

Minimum weight cycle basis problem is defined as the problem of finding an unoriented fundamental cycle matrix in which the total length of cycles is minimum. For graphs with positive arc weights, a solution can be found in polynomial time [22]. Here, our graph arc weights are 0 and 1. This algorithm generates a set of fundamental cycles, but restricts the generated cycles to a small set of $\mathcal{O}(nm)$ cycles, called Horton cycles. Each shortest path tree of the given graph has a set of fundamental cycles. The bound on Horton cycle is defined as m fundamental cycles of n shortest path trees. Every cycle in the minimum weight cycle basis is a Horton cycle [22]. Dijkstra's algorithm finds n shortest path trees and the Gaussian elimination is used to find independent cycles on an increasing ordered set of cycles. Figures 1, 3 and 4 depict graphs with their minimum weight cycle basis highlighted. Improvement for worst case time complexity of this algorithm was provided for undirected graphs [23], and planar graphs [24].