Cycle Flow Formulation of Optimal Network Flow Problems and Respective Distributed Solutions

Reza Asadi and Solmaz S. Kia, *Member, IEEE*

***Abstract*—In this paper, we use the cycle basis from graph theory to reduce the size of the decision variable space of optimal network flow problems by eliminating the aggregated flow conservation constraint. We use a minimum cost flow problem and an optimal power flow problem with generation and storage at the nodes to demonstrate our decision variable reduction method. The main advantage of the proposed technique is that it retains the natural sparse/decomposable structure of network flow problems. As such, the reformulated problems are still amenable to distributed solutions. We demonstrate this by proposing a distributed alternating direction method of multipliers (ADMM) solution for a minimum cost flow problem. We also show that the communication cost of the distributed ADMM algorithm for our proposed cycle-based formulation of the minimum cost flow problem is lower than that of a distributed ADMM algorithm for the original arc-based formulation.**

***Index Terms*—ADMM, cycle basis, distributed optimization, optimal network Flow.**

I. Introduction

In a network flow problem, a physical system consisting of several routes between source and sink points transfers input flows from the source points to the sink points. The primary objective of optimal network flow problems is to minimize the overall cost of transporting flow [1]. Network flow problems appear in many important applications, such as software-defined networking [2], wireless sensor networks [3], transportation systems [4], [5], and power networks [6]–[8]. In power network problems, variants of optimal network flow problems also aim to include optimal generation and storage costs [9]–[12].

With the advent of new technologies, the amount of available data and network size has been increasing, which necessitate various performance improvement techniques in cyber-physical systems [13], [14], and increase the size of optimization problems. However, the number of decision variables is directly related with the time and space (resources[1]) computation complexity of optimization solvers. Decision variable reduction techniques, as well as parallel/distributed optimization solutions, are therefore investigated to manage the complexity of large-scale optimization problems. Distributed solutions are of particular interest in decentralized cyber-physical operations, which aim to solve network flow problems in a scalable, fast, and decentralized manner. Although the computational cost of distributed algorithms in an optimal network flow problem is distributed among the cyber-layer nodes, the high number of decision variables normally translates to a large number of cyber nodes or large in-network communication overhead. Additionally limitations of these solutions include network congestion and the direct effect of communication cost on life-expectancy of battery-operated cyber nodes. To address these limitations, our objective is to construct a variable reduction technique that reduces the size of the decision variables in optimal network flow problems, while maintaining a sparse structure amenable to distributed solutions at a lower communication cost.

Variable fixing techniques [15], dominance techniques [16], and constraint pairing techniques [17] are general variable reduction techniques in integer quadratic problems. Also, in multi-objective optimization problems data mining techniques are used to reduce less effective variables [18]. For evolutionary optimization problems, [19] 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 for reducing the number of the search variables of the problem (see [20])

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

where $\phi:\mathbb{R}^m\to\mathbb{R}$ and $g:\mathbb{R}^m\to\mathbb{R}^r$ are the cost function and the inequality constraint function, respectively, and $\mathbf{A}\in\mathbb{R}^{n\times m}$ satisfies $\text{rank}(\mathbf{A})=\rho\le n<m$.

The affine feasible set for this optimization problem can be characterized as

$$\{\mathbf{x}\in\mathbb{R}^m \mid \mathbf{A}\mathbf{x}=\mathbf{b}\} = \{\mathbf{F}\mathbf{z}+\mathbf{x}^{\text{p}} \mid \mathbf{z}\in\mathbb{R}^{m-\rho}\}, \tag{2}$$

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

Manuscript received May 6, 2019; accepted July 11, 2019. This work was supported by National Science Foundation award ECCS-1653838. Recommended by Associate Editor Hongyi Li. *(Corresponding author: Reza Asadi.)*

Citation: R. Asadi and S. S. Kia, "Cycle flow formulation of optimal network flow problems and respective distributed solutions," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 5, pp. 1251–1260, Sept. 2019.

Reza Asadi is with the Computer Science Department, University of California, Irvine, CA 92697 USA (e-mail: rasadi@uci.edu).

Solmaz S. Kia is with the Mechanical and Aerospace Engineering Department, University of California, Irvine, CA 92697 USA (e-mail: solmaz@uci.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2019.1911705

[1] Resources are generically categorized as CPUs (number of independent computational units), primary storage (amount of memory/RAM), secondary storage (disk, cloud, etc.).

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

Compared to (1), in (3), the equality constraint is eliminated and the number of the decision variables are reduced from m to $m-\rho$.

Optimal network flow problems are normally in the form of the optimization problem (1), where the cost is the sum of the convex cost of the flow through the arcs subject to capacity bounds for each arc and flow conservation equations at each node. 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, and the constraints can be expanded to include other components. Nevertheless, an affine equality constraint that always is present in network flow problems is the flow conservation equation. The coefficient matrix of the linear equation describing the flow conservation constraints has a nullity of $m-n+1$. Therefore, one can employ the affine equality elimination method (2) to reduce the decision variables of the optimal flow problems by $\rho=m-n+1$ counts. This variable reduction can be substantial, since in network flow problems, the number of arcs m is usually much larger than the number of the nodes n. In time-varying problems where a network flow problem is solved over a time horizon of length N, the variable reduction is of $N(m-n+1)$ counts. However, the lack of efficient methods to construct matrix $\mathbf{F}$ and particular solution $\mathbf{x}^{\mathrm{p}}$ in (2) can be an impediment when using the affine equality constraint elimination method.

Various matrix factorization techniques, including LU, QR, LQ, SVD, and Gauss-Jordan elimination are proposed to compute the span of the null-space of a matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$ [21]–[23]. However, such techniques result in either a full matrix $\mathbf{F}$, which destroys the sparsity of the constraints and separability of the cost function of the optimal network flow problems, or a sparse $\mathbf{F}$, whose sparsity is not aligned with the physical layer of the network, and as a result produces an equivalent optimization problem that is hard to decentralize. Distributed solutions for network flow problems, e.g., in [1], [13], [24], [25] all take advantage of the sparsity of the flow conservation equation and separability of the cost function equations of the optimal network flow formulations in their developments. Moreover, arc-based DC and AC power flow with ADMM is studied in [26], [27].

Statement of Contribution: In this paper, we use a graph theoretic approach to reduce the decision variables of optimal network flow problems by eliminating the flow conservation constraint. In particular, we show that all solutions of the flow conservation equation are characterized explicitly in terms of the span of the columns of the transpose of a cycle basis matrix of the oriented network plus a particular solution. Cycle basis of a graph can be computed in polynomial time using efficient algorithms such as those in [28], [29]. To compute a particular solution, we then propose a graph theoretic approach. 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 single common sink node. An advantage of our proposed variable reduction method is that, for fixed physical layer networks, we only need to compute the cycle basis matrix and the fundamental particular solutions once. That is, the constituting components of our method are independent of the numerical values of the cost, the arc capacities or the input/output flow values. We demonstrate the application of our flow conservation equation elimination over a minimum cost flow problem (i.e., a static optimization problem), as well as an optimal power flow problem with storage and generation at the nodes (i.e., a time-varying optimization problem). Our next contribution is to show that our proposed cycle basis reduced-decision-variable formulation is amenable to distributed solutions–this is due to the sparse and particular composition of the cycle basis matrices that result in a tangible and topologically traceable cycle flow definition for the variables in the reduced space. In this regard, we demonstrate an implantation of a distributed ADMM solution method (see [30] and [31]) for our reduced-variable reformulation of the minimum cost network flow problem. To implement this distributed solution, we propose a cyber-layer whose nodes are defined based on the cycles of the cycle basis of the physical-layer graph, used in the variable reduction stage. This architecture is of particular interest for physical networks with articulation points[2]. For such networks, we show that the minimum cost optimal network flow problem decomposes into two or more decoupled smaller optimization problems, and each can be solved in a distributed manner independent from the others. Our final contribution is to show that distributed solutions for reduced-decision-variable reformulations can also be developed for conventional node-based cyber-layers. We use an example to show that the communication cost of the distributed ADMM algorithm for our proposed cycle-based formulation of the minimum cost flow problem is lower than that of a distributed ADMM algorithm for the original arc-based formulation. A preliminary version of parts of our results in this paper has appeared in [33].

Notations: $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\ge0}$, and $\mathbb{R}_{\le0}$ denote the set of real, positive real, non-negative real, and non-positive real numbers, respectively. $\mathbf{1}_n$ is the vector of n ones. $\mathbf{A}^{\top}$ is the transpose of a matrix $\mathbf{A}$ and $[\mathbf{A}]_i$ indicate its *ith* column. $[\{z_k\}_{k=1}^{n}]$ is 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, the aggregate vector of these variables is $\mathbf{p}=[\{p_i\}_{i=1}^{N}]\in\mathbb{R}^{N}$. The relative complement of a set $\mathcal{B}$ in a set $\mathcal{A}$ is $\mathcal{A}\backslash\mathcal{B}=\{x\in\mathcal{A}\,|\,x\notin\mathcal{B}\}$.

Organization of the Paper: Section II defines graph-related terminologies and some of the basic properties of the cycle basis method. Section III formally presents our decision variable reduction method and its application in a minimum cost flow and an optimal power flow problems. Section IV presents a distributed ADMM solution for the cycle-flow-

[2] An articulation point of an undirected connected graph is a node whose removal along with its incident edges disconnects the graph [32].

formulated minimum cost network flow problem and discusses the communication reduction on some example networks. Section V demonstrates the performance of the distributed cycle basis ADMM on a numerical example in terms of convergence to optimum solution and the number of communications. Section VI provides the conclusions.

II. A Review of Cycle Basis in Graphs

In this section, following [34], we review our graph-related terminology and conventions, and introduce our graph-related notations. We represent a graph of n nodes and m arcs with $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{v_1,v_2,\dots,v_n\}$ is the node set and $\mathcal{E}=\{e_1,\dots,e_m\}\in\mathcal{V}\times\mathcal{V}$ 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 $\mathcal{I}^{\mathrm{o}}\in\mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, where $\mathcal{I}^{\mathrm{o}}_{ij}=1$ if arc e_j leaves node v_i, $\mathcal{I}^{\mathrm{o}}_{ij}=-1$ if arc e_j enters node v_i, otherwise $\mathcal{I}^{\mathrm{o}}_{ij}=0$. For a connected graph of n nodes with a given orientation, the rank of $\mathcal{I}^{\mathrm{o}}$ 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 in the oriented graph $\mathcal{G}^{\mathrm{o}}$ is a cycle in the underlying undirected graph $\mathcal{G}$. We assign the counter clockwise direction as positive cycles orientation. Associated with each cycle is a *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. The following relationship exists between every oriented cycle vector and the oriented incidence matrix.

Lemma 1 (Relationship between the oriented incidence matrix and an oriented cycle vector (see [34] for proof)): In an oriented graph $\mathcal{G}^o$, every oriented cycle vector $\mathbf{c}^o$ is orthogonal to every row of oriented incident matrix $\mathcal{I}^o$, i.e., $\mathcal{I}^o\,\mathbf{c}^o=\mathbf{0}_n$. ■

Next, note that the vector space over Q, the Galois field of 2, generated by the oriented cycle vectors is the cycle space of $\mathcal{G}^{\mathrm{o}}$. A set of cycles is called a *cycle basis* if it forms a basis for this vector space. When $\mathcal{G}$ is connected, the cycle space has dimension $\mu=m-n+1$. Given a cycle basis, we define an *oriented cycle basis matrix* $\mathbf{B}^{\mathrm{o}}\in\mathbb{R}^{\mu\times m}$ as a matrix whose rows are each the transpose of the oriented cycle vector of the cycles of this cycle basis. This matrix satisfies $\operatorname{rank}(\mathbf{B}^{\mathrm{o}})=m-n+1$. We associate a weight 1 to each arc of $\mathcal{G}^{\mathrm{o}}$. The weight of a cycle basis is the sum of the weights of its cycles. A *minimum cycle basis of* $\mathcal{G}^{\mathrm{o}}$ is a cycle basis of minimum weight.

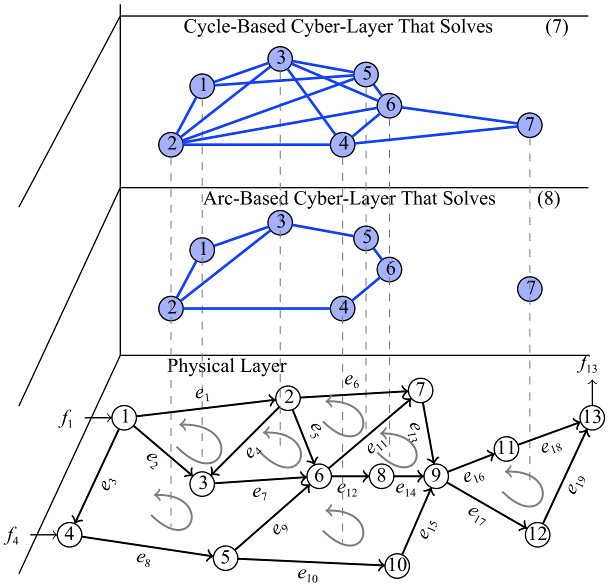

Fig. 1. A physical network with two cycle-based cyber-layer network overlaid atop. The physical layer network has two source nodes v_1 and v_4 and one sink node v_{13}. The physical layer has 19 arcs and 7 cycles. The lower cyber-layer belongs to a case that cyber-layer nodes are solving minimum network flow problem (8) using our proposed cycle-flow based distributed ADMM algorithm. The top cyber-layer belongs to a case that cyber-layer nodes are solving problem (7) using an arc-flow based distributed ADMM algorithm proposed in [24]. The connection lines in each cyber-layer are based on the variable dependency between the cyber nodes.

III. Decision Variable Reduction in Optimal 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 with an arc. For example, in a power network the arc is a transmission line, while in a transportation network the arc 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 arc, however, we assume a pre-specified positive orientation for each arc and based on it we describe the flow network in the physical layer by the oriented version of $\mathcal{G}_{\text{physic}}$, i.e., $\mathcal{G}^{\mathrm{o}}_{\text{physic}}=(\mathcal{V}_{\text{physic}},\mathcal{E}^{\mathrm{o}}_{\text{physic}})$. This physical network transfer flow(s) from a set of source nodes to a set of sink nodes (see physical layer in Fig. 1), 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. Given x_i be the flow across the arc $e_i\in\mathcal{E}^{\mathrm{o}}_{\text{physic}}=\{e_1,e_2,\dots,e_m\}$, every arc $e_i\in\mathcal{E}^{\mathrm{o}}_{\text{physic}}$ has a pre-specified capacity, i.e., $\mathrm{b}_i\le x_i\le\mathrm{c}_i$, for some known $\mathrm{b}_i,\mathrm{c}_i\in\mathbb{R}$. For any external flow f_i, we use the sign convention of $\mathrm{f}_i>0$ for input flow and $\mathrm{f}_i<0$ for output flow, and $\mathrm{f}_i=0$ otherwise. Using the oriented incidence matrix, the conservation of the flow constraint at each node can be represented as

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

which in an aggregated form is represented with

$$\mathcal{I}^{\mathrm{o}} \mathbf{x} = \mathbf{f}, \tag{5}$$

where $\mathbf{f} = (\mathrm{f}_1,\cdots,\mathrm{f}_n)^\top$. Note that since $\mathrm{rank}(\mathcal{I}^{\mathrm{o}}) = n-1$ and $\mathbf{1}_n^\top \mathcal{I}^{\mathrm{o}} = \mathbf{0}$, every set of valid exogenous input/output flow should satisfy $\sum_{i=1}^{n} \mathrm{f}_i = 0$. The following result uses Lemma 1, to characterize the solutions of (5).

Theorem 1 (Solution set of (5)): *Let* $\sum_{i=1}^{n} \mathrm{f}_i = 0$. Then, the solution set of the aggregated conservation of flow (5) is given by

$$\{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{x} = \mathbf{B}^{\mathrm{o}\,\top}\mathbf{z} + \mathbf{x}^{\mathrm{p}}, \quad \mathbf{z} \in^{m-n+1}\}, \tag{6}$$

where $\mathbf{x}^{\mathrm{p}}$ is a particular solution of (5).

Proof: The proof relies on showing that the null-space of $\mathcal{I}^{\mathrm{o}}$ is spanned by columns of $\mathbf{B}^{\mathrm{o}\,\top}$. By virtue of Lemma 1, we have $\mathcal{I}^{\mathrm{o}}\mathbf{B}^{\mathrm{o}\,\top} = \mathbf{0}$. Recall that for a connected oriented graph $\mathrm{rank}(\mathcal{I}^{\mathrm{o}}) = n-1$. Because $\mathrm{rank}(\mathbf{B}^{\mathrm{o}\,\top}) = m-n+1$, null-space of $\mathcal{I}^{\mathrm{o}} \in \mathbb{R}^{n\times m}$ is spanned by columns of $\mathbf{B}^{\mathrm{o}\,\top}$. ■

In what follows, we use the result of Theorem 1 to develop an affine equality elimination method to reduce the decision variables of two optimal network flow problems. However, the effectiveness of this approach depends on how efficiently one can construct matrix $\mathbf{B}^{\mathrm{o}}$ and particular solution $\mathbf{x}^{\mathrm{p}}$, especially in large scale networks. In regards to $\mathbf{B}^{\mathrm{o}}$, there are several efficient polynomial time algorithms in the literature (see e.g., [28], [29]); see Section V for the cycle matrix corresponding to the minimum weight cycle basis of the network of Fig. 1.

Next, we propose a simple method to construct a particular solution $\mathbf{x}^{\mathrm{p}}$ 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 $\mathrm{f}_i = 1$ entering the network at node v_i and leaving it at node v_j can simply be constructed by assuming that $\mathrm{f}_i = 1$ flows along a path from node v_i to node v_j.

Lemma 2: (Particular solution of (5)). Given an input/output flow vector $\mathbf{f}$ over $\mathcal{G}^o_{physic}$ which satisfies $\sum_{i=1}^{n} \mathrm{f}_i = 0$, a particular solution for (5) is $\mathbf{x}^p = \sum_{i=1}^{n-1} \mathrm{f}_i \bar{\mathbf{x}}^{p,v_i}$. Here, $\bar{\mathbf{x}}^{p,v_i} \in \mathbb{R}^m$, $i \in \{1,\ldots,n-1\}$, is constructed from a path that connects node v_i to node v_n such that $\mathrm{x}_j^{p,v_i} = 1$ (resp. $\mathrm{x}_j^{p,v_i} = -1$) if e_j, $j \in \{1,\ldots,m\}$, is on this path and is along (resp. opposing) the direction of the path, otherwise $\mathrm{x}_j^{p,v_i} = 0$.

Proof: For every $i \in \{1,\ldots,n-1\}$, consider a virtual scenario where a unit flow $\bar{\mathrm{f}}_i = 1$ 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 $\mathcal{I}^{\mathrm{o}}\bar{\mathbf{x}}^{\mathrm{p},v_i} = \bar{\mathbf{f}}^{v_i}$, $i \in \{1,\ldots,n-1\}$ where $\bar{\mathrm{f}}_i^{v_i} = 1$, $\bar{\mathrm{f}}_n^{v_i} = -1$ and $\bar{\mathrm{f}}_j^{v_i} = 0$, $j \in \{1,\ldots,n\}\backslash\{i,n\}$. For a given network flow vector $\mathbf{f}$ because $\mathrm{f}_n = -\sum_{i=1}^{n-1}\mathrm{f}_i$, we can write $\mathbf{f} = \sum_{i=1}^{n-1}\mathrm{f}_i\bar{\mathbf{f}}^{v_i}$. Therefore, $\mathcal{I}^{\mathrm{o}}\sum_{i=1}^{n-1}\mathrm{f}_i\bar{\mathbf{x}}^{\mathrm{p}}_{v_i} = \sum_{i=1}^{n-1}\mathrm{f}_i\bar{\mathbf{f}}^{v_i} = \mathbf{f}$. ■

A few remarks are in order regarding the particular solution. First, note that construction of the `fundamental' particular solution set $\{\bar{\mathbf{x}}^{\mathrm{p},v_i}\}_{i=1}^{n-1}$ is independent 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. Finally, to obtain sparse elementary solutions, we can use a shortest path between nodes v_i and v_n. A numerical example discussing the particular solution of the flow conservation equation for the network of Fig. 1 is given in Section V.

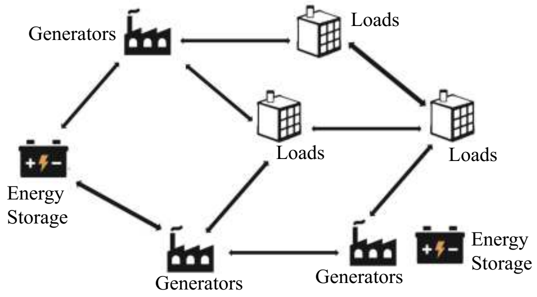

Fig. 2. A schematic representation of a network flow problem with generators and energy storage and loads in a network.

A. Minimum Cost Flow Problem

We consider a minimum cost flow problem over $\mathcal{G}^{\mathrm{o}}_{\mathrm{physic}}$ 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 $e_i \in \mathcal{E}^o_{\mathrm{physic}}$, and our objective is to find the network minimizer $\mathbf{x}^\star \in \mathbb{R}^m$ in the following optimization problem

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathbb{R}^m}{\mathrm{argmin}}\; \phi(\mathbf{x}) = \sum\nolimits_{i=1}^{m} \phi_i(x_i), \;\; \text{s.t.} \tag{7a}$$

$$\sum\nolimits_{j=1}^{m} \mathcal{I}^{\mathrm{o}}_{ij} x_j = \mathrm{f}_i; \quad i \in \{1,\ldots,n\}; \tag{7b}$$

$$\mathrm{b}_j \le x_j \le \mathrm{c}_j; \qquad j \in \{1,\cdots,m\}, \tag{7c}$$

where $\mathbf{f} = (\mathrm{f}_1,\cdots,\mathrm{f}_n)^\top$ is the given input/output flow vector which satisfies $\sum_{i=1}^{n}\mathrm{f}_i = 0$. Here, (7b) captures the flow conservation at nodes across the network and (7c) describes the arc capacity constraints. The number of search variables in the optimization problem (7) is equal to the number of the arcs of the network, i.e., $|\mathcal{E}^o_{\mathrm{physic}}| = m$. Our result below uses Theorem 1 to eliminate the affine equality constrains (7b) and reduce the search variables to $m-n+1$.

Theorem 2: (Eliminating the flow conservation constraint from (7)). Consider the optimal network flow problem (7) over a connected physical network $\mathcal{G}^o_{\mathrm{physic}}$. Then, $\mathbf{x}^\star$ in (1) satisfies $\mathbf{x}^\star = \mathbf{B}^{\mathrm{o}\,\top}\mathbf{z}^\star + \mathbf{x}^{\mathrm{p}}$ where

$$\mathbf{z}^\star = \underset{\mathbf{z}\in\mathbb{R}^{m-n+1}}{\mathrm{argmin}}\, \phi(\mathbf{z}) = \sum_{i=1}^{m} \phi_i(\mathbf{z}^\top[\mathbf{B}^{\mathrm{o}}]_i + \mathrm{x}_i^p), \;\; \text{s.t.}$$
$$\mathrm{b}_j \le \mathbf{z}^\top[\mathbf{B}^{\mathrm{o}}]_j + \mathrm{x}_j^p \le \mathrm{c}_j, \quad j \in \{1,\ldots,m\}, \tag{8}$$

and $\mathbf{B}^{\mathrm{o}}$ is an oriented cycle basis matrix of $\mathcal{G}^o_{physic}$ and $\mathbf{x}^{\mathrm{p}}$ is a particular solution of $\mathcal{I}^o\mathbf{x} = \mathbf{f}$.

Proof: The proof follows from invoking the same argument that is used to relate solutions of the optimization problem (1) to those of (3), and the result of Theorem 1. ■

B. Optimal Power Flow With Storage and Generation at Nodes

Next, we consider an optimal power flow problem over a network described by $\mathcal{G}^{\mathrm{o}}_{\mathrm{physic}}$ 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 $\mathcal{T}=\{1,\dots,T\}$. Mathematical modeling of this problem over various scenarios, including deterministic and stochastic generators, is considered in the literature [9]–[12].

All of 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 [9], which states the problem as a direct current (DC) power flow problem (see (9)). Without loss of generality, let at each time $t\in\mathcal{T}$, each node $v_i\in\mathcal{V}_{\mathrm{physic}}$ have a generator that 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 $\mathrm{d}_i(t)\in\mathbb{R}_{\leq 0}$. If a node does not have any of the generation, storage, or load components, the respective variables should simply be removed from the formulation below. Then, given a known load profile $\{\mathbf{d}(t)\}_{t=1}^{T}$, where $\mathbf{d}(t)=[\{\mathrm{d}_i(t)\}_{i=1}^{n}]$, the optimization problem of interest in [9] is

$$\{\mathbf{x}^\star(t),\boldsymbol{\delta}^\star(t),\mathbf{u}^\star(t),\mathbf{s}^\star(t),\boldsymbol{\theta}^\star(t)\}_{t=1}^{T}= \tag{9a}$$

$$\operatorname{argmin}\frac{1}{T}\sum\nolimits_{t=1}^{T}\Big(\sum\nolimits_{j=1}^{n}g_j(\delta_j(t))+\sum\nolimits_{i=1}^{m}\phi_i(x_i(t))\Big), \tag{9b}$$

$$\text{s.t. for } t\in\mathcal{T},\ \ i\in\{1,\dots,n\}\sum\nolimits_{j=1}^{m}\mathcal{I}^{\mathrm{o}}_{ij}x_j(t)=\delta_i(t)-u_i(t)+\mathrm{d}_i(t); \tag{9c}$$

$$s_i(t+1)=\lambda_i s_i(t)+u_i(t); \tag{9d}$$

$$\mathbf{B}_{ik}(\theta_i(t)-\theta_k(t))=x_j(t);\quad k\in\mathcal{N}^e(i),e_j=(v_i,v_k) \tag{9e}$$

$$\underline{\delta}_i\leq\delta_i(t)\leq\bar{\delta}_i;\ \ \underline{\mathrm{u}}_i\leq u_i(t)\leq\bar{\mathrm{u}}_i;\ \ \underline{\mathrm{s}}_i\leq s_i(t)\leq\bar{\mathrm{s}}_i; \tag{9f}$$

$$\mathrm{b}_j\leq x_j(t)\leq\mathrm{c}_j;\qquad j\in\{1,\dots,m\}, \tag{9g}$$

where the cost function includes generator costs g_j, $j\in\{1,\dots,n\}$, and transmission cost ϕ_i, $i\in\{1,\dots,m\}$, following DC power grid model in [35]. 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, $(\underline{\mathrm{s}}_i,\bar{\mathrm{s}}_i)\in\mathbb{R}\times\mathbb{R}$, $(\underline{\mathrm{u}}_i,\bar{\mathrm{u}}_i)\in\mathbb{R}\times\mathbb{R}$, $(\underline{\delta}_i,\bar{\delta}_i)\in\mathbb{R}\times\mathbb{R}$, are known as lower bound and upper bound values on storage level, battery charge/discharge, and power generation by the generator, respectively. Finally $\mathbf{B}_{ik}(\theta_i(t)-\theta_k(t))$ is the DC approximation for alternating current power flow. Here, $\theta_i(t)$ is the voltage phase angle of node (bus) $v_i\in\mathcal{V}_{\mathrm{physic}}$ at time t and $\mathbf{B}\in\mathbb{R}^{n\times n}$ is the imaginary part of the admittance matrix under DC assumption (for more details see [9]).

The following result shows that the number of search variables related to the flow in the optimization problem (9) 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 (9c), we need to introduce a new set of affine equality constraint (10b) which ensure balance between the external input and output flows.

Proposition 1: (Eliminating the flow conservation constraint from (9)). Consider the optimal power flow problem (9) over a physical network described by $\mathcal{G}^{o}_{\mathrm{physic}}$ with a given set of loads $\{\mathbf{d}(t)\}_{t=1}^{T}$. Then, $\{\mathbf{B}^{\mathrm{o}\,\top}\mathbf{z}^\star(t)+\mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta}^\star(t),\mathbf{u}^\star(t),\mathbf{d}(t)),\mathbf{u}^\star(t),\mathbf{s}^\star(t),\boldsymbol{\theta}^\star(t)\}_{t=1}^{T}$ is a minimizer of the optimization problem (9) where

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

$$\operatorname{argmin}\frac{1}{T}\sum\nolimits_{t=1}^{T}\Big(\sum\nolimits_{j=1}^{n}g_j(\delta_j(t))+ \tag{10a}$$

$$\sum\nolimits_{i=1}^{m}\phi_i(\mathbf{z}(t)^\top[\mathbf{B}^{\mathrm{o}}]_i+\mathrm{x}_i^{p}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t))\Big),\ \textit{s.t.}$$

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

$$\sum\nolimits_{j=1}^{n}(\delta_j(t)+u_j(t)+\mathrm{d}_j(t))=0, \tag{10b}$$

$$s_i(t+1)=\lambda_i s_i(t)+u_i(t), \tag{10c}$$

$$\begin{aligned}\mathbf{B}_{ik}(\theta_i(t)-\theta_k(t))&=\mathbf{z}(t)^\top[\mathbf{B}^{\mathrm{o}}]_j+\mathrm{x}_j^{p}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t)),\\ &k\in\mathcal{N}^e(i),e_j=(v_i,v_k),\end{aligned} \tag{10d}$$

$$\underline{\delta}_i\leq\delta_i(t)\leq\bar{\delta}_i,\ \ \underline{\mathrm{u}}_i\leq u_i(t)\leq\bar{\mathrm{u}}_i,\ \ \underline{\mathrm{s}}_i\leq s_i(t)\leq\bar{\mathrm{s}}_i, \tag{10e}$$

$$\begin{aligned}\mathrm{b}_j\leq\mathbf{z}(t)^\top[\mathbf{B}^{\mathrm{o}}]_j+\mathrm{x}_j^{p}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t))\leq\mathbf{c}_j,&\\ j\in\{1,\cdots,m\}.&\end{aligned} \tag{10f}$$

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

Proof: The equality constraint (9c) in aggregated form is

$$\mathcal{I}^{\mathrm{o}}\mathbf{x}(t)=\boldsymbol{\delta}(t)+\mathbf{u}(t)+\mathbf{d}(t),\qquad t\in\mathcal{T}. \tag{11}$$

Note that rank of $\mathcal{I}^{\mathrm{o}}\in\mathbb{R}^{n\times m}$ is $n-1$ and $\mathbf{1}_n^\top\mathcal{I}^{\mathrm{o}}=\mathbf{0}$, where $\mathbf{1}_n$ is the vector of n ones. Left multiplying (11) by $\mathbf{1}_n^\top$ results in (10b). Then, for a given $\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t)\in\mathbb{R}^n$ that satisfy (10b), following the method discussed in Lemma 1, we can show that $\mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t))=\sum_{i=1}^{n-1}(\delta_i(t)+u_i(t)+\mathrm{d}_i(t))\,\bar{\mathbf{x}}^{\mathrm{p},v_i}$ is a particular solution of (11), i.e., $\mathcal{I}^{\mathrm{o}}\mathbf{x}^{\mathrm{p}}(\boldsymbol{\delta}(t),\mathbf{u}(t),\mathbf{d}(t))=\boldsymbol{\delta}(t)+\mathbf{u}(t)+\mathbf{d}(t)$. Therefore, for any given load vector $\mathbf{d}$ (to simplify the notation we drop argument t), we have

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

As a result, at each $t\in\mathcal{T}$, we can eliminate the affine equation (9c) and obtain the equivalent optimization problem (10). As it is described in the statement, the minimizers of problem (10) and (9) are equal. ■

IV. A Cycle-Basis Distributed ADMM Algorithm for Minimum Cost Network Flow Problem

In this section, our objective is to demonstrate the effectiveness of the cycle basis decision variable reduction technique in decreasing the communication cost of distributed solutions of optimal network flow problems. We use the minimum network flow problem (7) as our demonstration case. For this problem, we first develop a distributed ADMM algorithm to solve (8). Then, we compare the communication cost of this algorithm to that of a distributed ADMM solution for (7).

To develop our results, we introduce first some notations related to the oriented cycles of $\mathcal{G}^{\mathrm{o}}_{\mathrm{physic}}$. Suppose $\mu = m-n+1$. Let $C=\{C_i\}_{i=1}^{\mu}$, be the set of cycles of corresponding to $\mathbf{B}^{\mathrm{o}}$ that is used to eliminate the flow conservation equation as explained in Section III. We represent the set of arcs of any $C_i \in C$, $i\in\{1,\dots,\mu\}$, by $\mathcal{E}^C_i = \{e_j \in \mathcal{E}^{\mathrm{o}}, j\in\{1,\dots,m\} \mid \mathbf{B}^{\mathrm{o}}_{ij} \neq 0\}$. For a given cycle basis C, we refer to the cycles that share an arc as neighbors and represent the set of neighboring cycles of any cycle $C_i\in C$, $i\in\{1,\dots,\mu\}$, by $\mathcal{N}^C_i = \{j\in\{1,\dots,\mu\}\backslash\{i\} \mid \exists k \in \{1,\dots,m\}$ s.t. $\mathbf{B}^{\mathrm{o}}_{ik}\neq 0$ and $\mathbf{B}^{\mathrm{o}}_{jk}\neq 0\}$.

We let $\mathcal{I}^{\mathrm{c}}(e_i)$ be the set of indexes of the cycles that arc $e_i\in\mathcal{E}_{\mathrm{physic}}$ belongs to them, i.e., $\mathcal{I}^{\mathrm{c}}(e_i)=\{j\in\{1,\dots,\mu\}\mid e_i\in\mathcal{E}^C_j\}$. We let $\mathcal{J}^{\mathcal{E}}(v_i)$ be the set of all the arcs that go through node $v_i\in\mathcal{V}_{\mathrm{physic}}$, i.e., $\mathcal{J}^{\mathcal{E}}(v_i)=\{v_j\in\mathcal{V}_{\mathrm{physic}}\backslash\{v_i\}\mid(v_i,v_j)\in\mathcal{E}_{\mathrm{physic}}\}$.

Recall that to eliminate the flow conservation equation we used $\mathbf{x}=\mathbf{B}^{\mathrm{o}\top}\mathbf{z}+\mathbf{x}^{\mathrm{p}}$, or equivalently $x_i=\mathbf{z}^\top[\mathbf{B}^{\mathrm{o}}]_i+\mathrm{x}^{\mathrm{p}}_i$, $i\in\{1,\dots,m\}$. Since, for a given arc $e_i\in\mathcal{E}^{\mathrm{o}}_{\mathrm{physic}}$, every element of $\mathbf{B}^{\mathrm{o}}_{ji}$ is zero except if cycle C^{of}_j contains arc e_i, every x_i, $i\in\{1,\dots,m\}$, is an affine function of $\mathrm{x}^{\mathrm{p}}_i$ and $\{z_k\}_{k\in\mathcal{I}^{\mathrm{c}}(e_i)}$. If we think of every z_i, $i\in\{1,\dots,\mu\}$ as a cycle flow variable (with positive direction in counterclockwise direction) of the cycle C_i, then every arc flow x_j, $j\in\{1,\dots,m\}$ is a function of its corresponding component of the particular solution and the cycle flows of the cycles that contain the arc. Given such relationship, the cost function of every arc is

$$\phi_i(x_i)=\phi_i(\mathbf{z}^\top[\mathbf{B}^{\mathrm{o}}]_i+\mathrm{x}^{\mathrm{p}}_i)=\psi_i(\{z_k\}_{k\in\mathcal{I}^{\mathrm{c}}(e_i)}).$$

Next, we derive equivalent representations of optimization problem (8) that can be solved in a distributed manner using the ADMM algorithm of [13] via two different cyber-layer architectures.

Cycle-Based Cyber-Layer: to develop our first distributed solution to solve (8), we assign a cyber-layer node to each cycle of the cycle basis that we used in our decision variable reduction stage (see Fig. 1 as an example). We refer to this architecture as *cycle-based cyber-layer*. We assume that the cyber-layer nodes of the neighboring cycles can communicate with each other in a bi-directional way. This procedure results in a cyber-layer with μ nodes. To obtain cycle basis with fewest number of arcs in each cycle we can use a minimum weight cycle basis algorithms (see e.g. [29]).

Now, for every cyber-layer node $i\in\{1,\dots,\mu\}$, we define $\mathbf{y}_i=(\bar{y}_i,\tilde{\mathbf{y}}_i)\in\mathbb{R}^{|\mathcal{N}^C_i|+1}$, where $\bar{y}_i\in\mathbb{R}$ is the local copy of z_i and $\tilde{\mathbf{y}}_i$ is the local copy of $\{z_k\}_{k\in\mathcal{N}^C_i}$ 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 each 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 $\{z_j\}_{j\in\mathcal{I}^{\mathrm{c}}(e_k)}$. For every cyber-layer node, we define its cost function as

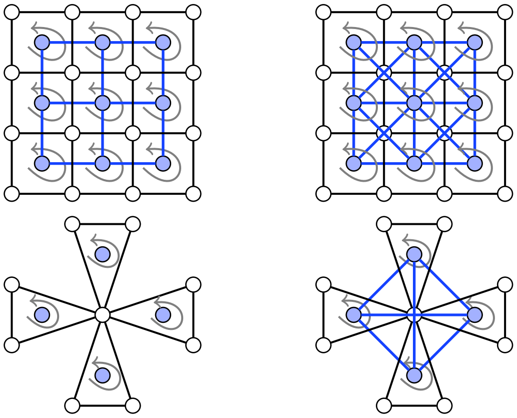

Fig. 3. Examples of physical layer networks (black network) with the corresponding cycle-based cyber-layer atop (blue thicker network).

$$\theta_i(\mathbf{y}_i)=\sum_{\forall e_k\in\mathcal{E}^C_i}\frac{1}{|\mathcal{I}^{\mathrm{c}}(e_i)|}\psi_k(\mathbf{y}_i(e_k)). \tag{12}$$

Then, we can cast the minimum cost network flow problem (8) in the following equivalent form

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

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

$$\begin{cases}\mathbf{y}_i(e_k)=[\{\bar{y}_j\}_{j\in\mathcal{I}^{\mathrm{c}}(e_i)}],\\ \mathrm{b}_k\leq\mathbf{y}_i(e_k)^\top[\{\mathbf{B}^{\mathrm{o}}_{jk}\}_{j\in\mathcal{I}^{\mathrm{c}}(e_k)}]+\mathrm{x}^{\mathrm{p}}_k\leq\mathrm{c}_k,\end{cases}\forall e_k\in\mathcal{E}^C_i. \tag{13}$$

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 (13) fits the standard framework developed for distributed ADMM solutions and can be solved for example using the algorithm of [13]. The details are omitted for brevity. Once every cyber node $i\in\{1,\dots,\mu\}$ computes $\mathbf{y}^\star_i$, then it can use $x^\star_k=\mathbf{y}^\star_i(e_k)^\top[\{\mathbf{B}^{\mathrm{o}}_{jk}\}_{j\in\mathcal{I}^{\mathrm{c}}(e_k)}]+\mathrm{x}^{\mathrm{p}}_k$ to obtain the optimal cycle flows through its arcs $e_k\in\mathcal{E}^C_i$. 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 large cycles, one can split the cycle among several cyber nodes. A numerical example demonstrating our proposed distributed ADMM algorithm over a cycle-based cyber-layer can be found in authors' preliminary work [33].

One can solve also the original minimum cost network flow problem (7) in a distributed manner by a cycle-based cyber-layer using, for example, the distributed ADMM algorithm of [24]. To do so, every cyber-layer node needs a local copy of the arc flows across all the nodes in its corresponding cycle. The local copies are required to create a local copy of the flow conservation equation of the nodes. Then, to coordinate these local copies, the cyber-layer agents that share a node are required to communicate with each other. Since there are more cycles that are connected to each other through the nodes than those connected by arcs, as shown in Fig. 1, this distributed solution requires more connections in the cyber-layer than the distributed solution for (8). Moreover, because there are more arc flows than cycle flows, the message sizes exchanged among neighboring cyber agents solving (7) is larger. Therefore, the distributed solution for (7) will be less favorable from network congestion and communication energy consumption perspective.

Remark 1: (Example networks). In a square mesh physical layer network with $n = N^2$ nodes, the total number of arcs is $m = 2(N^2 - N)$, and the total number of cycles is $c = N^2 - 2N + 1$, see the top plots in Fig. 3. In the mesh networks of Fig. 3, cyber-layer constructed based on the cycle-basis partitioning of the physical layer is shown via the thick blue network. For a cyber-layer which solves the alternative optimal network flow problem (8) via an ADMM distributed solution, a cyber agent only needs to communicate with neighboring agents which have a common arc. The number of communication links in cyber-layer is $m_1 = 2((N-1)^2 - (N-1)) = 2(N^2 - 3N + 2)$. For a cyber-layer which solves the original optimal network flow problem (7) via an ADMM distributed solution, any cyber agent needs to communicate with neighboring agents which has a common node in the graph. Therefore, the number of communication links in cyber-layer is $m_2 = ((N-1)^2 - 2(N-1) + 1)2 + 2(N^2 - 3N + 2) = 4N^2 - 14N + 12$. Consequently, we can obtain that $\lim_{n\to\infty} \frac{m_2}{m_1} = 2$. An interesting case involving our proposed cycle-based distributed ADMM solution of (13) is when the physical layer network has articulation points (e.g., node 27 in IEEE bus system 30 in Fig. 4 and node 9 of the physical layer in Fig. 1 are articulated points). The physical layer graph with an articulation point is consisted of subgraphs that are connected to each other through simple nodes with no common arcs between them. Therefore, when our proposed aforementioned ADMM algorithm is implemented, there is no variable dependency between the cyber nodes of these subgraphs. This means that the original fully coupled optimization problem is now divided into smaller optimization problems that each can be solved independently. For example in the network of Fig. 1, cyber-layer node 1 can obtain the optimal solution across its arcs in one ADMM iteration without a need to coordinate with other cyber agents. An extreme example case is also shown in the bottom plots of Fig. 3. In this example there is no common arc among cycles, but they all have one common node. The number of communication links for cycle-based cyber-layer graph is zero. In this case, a cycle-based ADMM algorithm finds the optimum solution in one step, without any communication. However, in arc-based network flow, the cyber-layer is a complete graph as all of the agents are connected to each other.

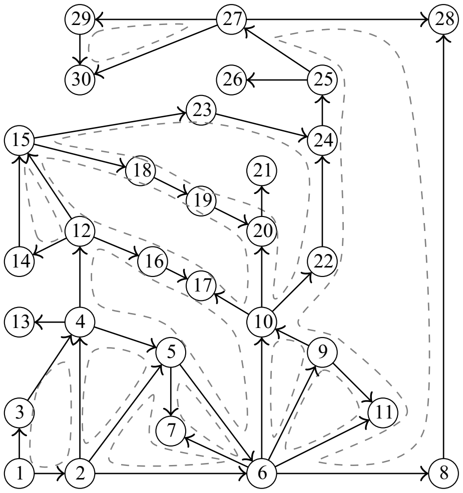

Fig. 4. The graph related to IEEE bus system 30 with 41 arcs, 30 nodes and 12 cycles.

Node-Based Cyber-Layer: We can solve problem (8) in a distributed manner with the conventional *node-based cyber-layer*, as well. In a node-based cyber-layer architecture, a cyber agent is assigned to each node of the physical layer to compute the flow across all the incident arcs of the corresponding physical layer node. For example, in traffic networks, a cyber agent is assigned to each intersection, which is a node in the physical layer. In this case, the topology of the cyber-layer is exactly the same as the physical layer. To solve problem (8) using a distributed ADMM algorithm over such cyber-layer, we assume that each cyber agent has a local variable $\mathbf{y}_i$ consisted of local copies of the cycle flows that goes across its arcs. We also split the flow cost across each arc between each end nodes. Then, the total local cost at each cyber node is

$$\theta_i(\mathbf{y}_i) = \frac{1}{2} \sum_{\forall e_k \in \mathcal{J}^{\mathcal{E}}(v_i)} \psi_k(\mathbf{y}_i(e_k)),\ v_i \in \mathcal{V}_{\text{physic}}. \tag{14}$$

Then, we can cast the minimum cost network flow problem (8) in the following equivalent form

$$\mathbf{y}^\star = \underset{\mathbf{y}_1,\cdots,\mathbf{y}_n}{\operatorname{argmin}} \sum\nolimits_{i=1}^{n} \theta_i(\mathbf{y}_i), \quad \text{s.t.}$$

Set of constraints at each cyber agent $i \in \{1,\ldots,n\}$:

$$\begin{cases} \mathbf{y}_i(e_k) = \mathbf{y}_j(e_k),\ e_k = (v_i, v_j) \in \mathcal{E}_{\text{physic}}, \\ \mathrm{b}_k \leq \mathbf{y}_i(e_k)^\top [\{\mathbf{B}^{\mathrm{o}}_{lk}\}_{l \in \mathcal{I}^{\mathrm{c}}(e_k)}] + \mathrm{x}^{\mathrm{p}}_k \leq \mathrm{c}_k, \end{cases} \forall e_k \in \mathcal{J}^{\mathcal{E}}(v_i), \tag{15}$$

Given this equivalent representation, similar to the method described for the cycle-based cyber-layer, we can now solve the optimization problem (8) using a distributed ADMM algorithm. In a cycle flow representation, it is more likely to have fewer cycles that go through a node than the arcs that are incident at that node. Therefore, in a distributed ADMM solution, the size of the broadcast messages of a cyber agent solving (8) is more likely to be less than the size of the broadcast messages when we solve (7).

V. Demonstrative Examples

We demonstrate the use of distributed cycle-flow-based ADMM algorithm for a minimum cost optimal flow over the network shown in Fig. 1, and compare it with the arc-flow-based ADMM distributed algorithm [24]. The arrows in the physical layer show the positive flow orientation assigned to the arcs. The cyber-layer is generated based in the minimum weight cycle basis partitioning of the physical layer as shown in the bolder network with blue nodes in Fig. 1. In this problem, we set capacity bounds $\mathrm{b}_i = -\mathrm{c}_i$, and $\mathrm{c}_i \in [1,20]$, $i \in \{1,\ldots,19\}$. We assume that the cost of the network flow at each arc is given as $\phi_i(x_i) = \frac{1}{2}\mathrm{a}_i x_i^2 + \mathrm{b}_i x_i$, where $x_i = \mathbf{z}^\top[\mathbf{B}^{\mathrm{o}}]_i + \mathrm{x}_i^{\mathrm{p}}$ is the arc flow and $\mathbf{z} = (z_1,\cdots,z_7)^\top$ are the cycle flows. The parameters of the cost functions are chosen randomly from $\mathrm{a}_i \in [1,20]$, $i \in \{1,\ldots,19\}$ and $\mathrm{b}_i \in [1,20]$, $i \in \{1,\ldots,19\}$. In the physical layer network in Fig. 1, there are two source nodes v_1 and v_4 and one sink node v_{13}. For our selected capacity bounds, using Edmonds-Krap algorithm [36] we obtain the maximum input flow $\mathrm{f}_1 + \mathrm{f}_4$ to be 30. In our simulation, then we set $\mathrm{f}_1 = 10$ and $\mathrm{f}_4 = 15$. The output follows are $\mathrm{f}_{13} = -25$. For the network of Fig. 1, there are 7 cycles and 19 edges in the network and the orientated cycle basis matrix is, (recall that we have assumed the positive cycle flow direction to be counterclockwise). We follow Lemma III.1 to generate a particular solution for given input output flow vector $\mathbf{f} = (\mathrm{f}_1, 0, 0, \mathrm{f}_4, 0, 0, 0, 0, 0, 0, 0, 0, \mathrm{f}_{13})^\top$, where $\mathrm{f}_{13} = -(\mathrm{f}_1 + \mathrm{f}_4)$ (recall that input flows have positive and output flows have negative values). We compute the elementary particular solutions for nodes v_1 and v_4 using shortest path from them to node v_{13}: $\bar{\mathbf{x}}^{\mathrm{p},v_1} = (1, \mathbf{0}_{1\times4}, 1, \mathbf{0}_{1\times6}, 1, 0, 0, 1, 0, 1, 0)^\top$, $\bar{\mathbf{x}}^{\mathrm{p},v_4} = (\mathbf{0}_{1\times7}, 1, 0, 1, \mathbf{0}_{1\times4}, 1, 1, 0, 1, 0)^\top$. Then $\bar{\mathbf{x}}^{\mathrm{p}} = \mathrm{f}_1\,\bar{\mathbf{x}}^{\mathrm{p},v_1} + \mathrm{f}_4\,\bar{\mathbf{x}}^{\mathrm{p},v_4}$.

$$\mathbf{B}^{\mathrm{o}} = \begin{bmatrix} -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 \end{bmatrix}$$

The central optimum solution of our example is $\mathbf{x}^\star = \{9.64, 2.35, –2, 0.03, 3.70, 5.90, 2.39, 13, 3, 10, 3.14, 5.94, 9.05, 5.94, 10, 14, 11, 14, 11\}$ with objective function value of $f^\star = 9316.8$ (also, $\mathbf{z}^\star = \{0.35, -2, 0.39, -5, 4.1, 0.95, 11\}$). Fig. 5 shows the absolute percentage error, defined as $\epsilon_f = (f^k - f^\star)/f^\star$ at iteration k of the cycle-flow-based and arc-flow-based distributed ADMM algorithms. The result shows that the convergence of the cycle-flow-based ADMM is faster than arc-flow-based ADMM when the problem is solved via the cycle-based cyber-layers shown in Fig. 1 for each of these approaches. To compare the number of communications between cycle-flow-based and arc-flow-based ADMM algorithms, we set a threshold of $\epsilon_f = 0.01$ for the absolute percentage error. Distributed cycle-flow-based ADMM converges after 5 itereation to $\epsilon_f = 0.01$ absolute percentage error while the distributed arc-flow-based ADMM converges after 13 iterations. Each iteration of the ADMM algorithm needs to send primal variables twice for every communication link (i,j) in cyber network from node i to node j and vice versa. The number of communication links in cycle-flow-based and arc-flow-based cyber-layer are 7 and 14 respectively. As a result, the number of communications for cycle-flow-based ADMM is 35 and for arc-flow-based ADMM is 112. In Fig. 6, we compare the number of communications between cycle-flow-based and arc-flow-based distributed ADMM for the cyber nodes shown in Fig. 1. The number of communications is obtained by adding up the communication incidences at each iteration, until we have $\|\mathbf{x}_i^k - \mathbf{x}_i^\star\| < \epsilon = 0.1$, where $\mathbf{x}_i^\star$ is the optimum solution of the central problem. Fig. 6 shows that the cycle-flow based algorithm requires fewer communications.

VI. Conclusion

We have considered optimal network flow problems and investigated how the decision variables size 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 the flow conservation equation in an efficient manner. In particular, we have shown that the components of our variable reduction method can be obtained in a systematic manner using graph theoretic approaches. Moreover, we have shown that the new formulation of the optimal network flow problems with reduced variables has a sparse structure and can be solved via distributed optimization solvers. In this regard, we have demonstrated the use of a distributed ADMM solver for the cycle-flow-based minimum cost flow formulation, and

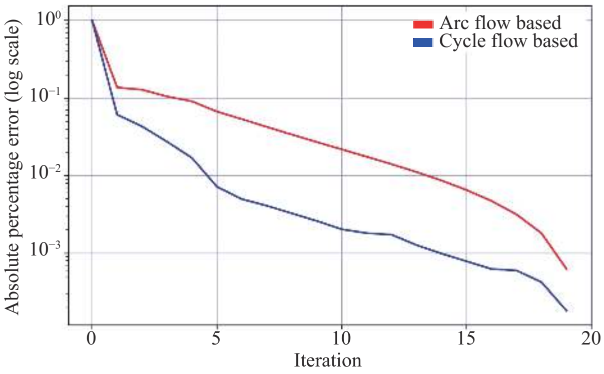

Fig. 5. $\epsilon_f = (f^k - f^\star)/f^\star$ versus iteration number k for the arc-flow-based and cycle-flow-based distributed ADMM algorithms over the cyber-layer architectures shown in Fig. 1.

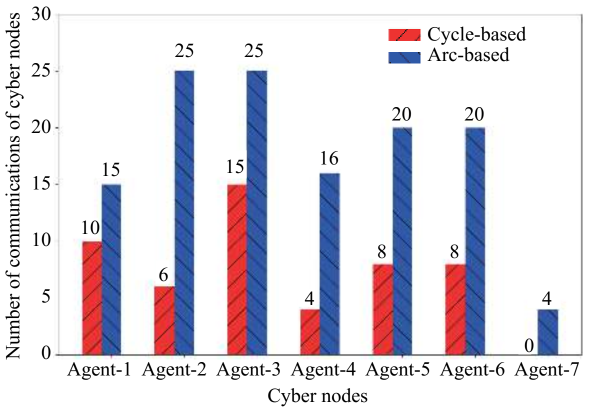

Fig. 6. Comparison of the number of communications between the cyber-layer nodes when they solve the optimal network flow problem using a distributed cycle-flow-based and arc-flow-based ADMM algorithms until we have $\|\mathbf{x}_i^k - \mathbf{x}_i^\star\| < \epsilon = 0.1$, $i \in \{1, \dots, 19\}$. As seen, in the arc-flow-based algorithms the cyber nodes communicate more.

showed that this distributed operation leads to a reduced communication cost among the cyber-layer nodes.

References

[1] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Citeseer, 1998.

[2] K. Qin, C. Huang, N. Ganesan, K. Liu, and X. Chen, "Minimum cost multi-path parallel transmission with delay constraint by extending openflow," *China Communications*, vol. 15, no. 3, pp. 15–26, 2018.

[3] A. Sinha and E. Modiano, "Optimal control for generalized networkflow problems," *IEEE/ACM Transactions on Networking* (*TON*), vol. 26, no. 1, pp. 506–519, 2018.

[4] C. Rosdahl, G. Nilsson, and G. Como, "On distributed optimal control of traffic flows in transportation networks," in *Proc. IEEE Conf. on Control Technology and Applications*, pp. 903–908, 2018.

[5] S. Pourazarm and C. G. Cassandras, "Optimal routing of energyaware vehicles in transportation networks with inhomogeneous charging nodes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2515–2527, 2018.

[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] C. D. Nicholson and W. Zhang, "Optimal network flow: a predictive analytics perspective on the fixed-charge network flow problem," *Computers & Industrial Engineering*, vol. 99, pp. 260–268, 2016.

[8] T. Soares, R. J. Bessa, P. Pinson, and H. Morais, "Active distribution grid management based on robust ac optimal power flow," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6229–6241, 2018.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] S. Rezaei, K. Kim, and E. Bozorgzadeh, "Scalable multi-queue data transfer scheme for fpga-based multi-accelerators," in *Proc. 2018 IEEE Int. Conf. on Computer Design* (*ICCD*), pp. 374–380.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] S. Boyd and L. Vandenberghe, *Convex optimization*. England, US: CUP, 2004.

[21] M. T. Heath, "Some extensions of an algorithm for sparse linear least squares problems," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 2, pp. 223–237, 1982.

[22] A. K. Cline and I. S. Dhillon, *Computation of the Singular Value Decomposition*. CRC Press, 2006.

[23] M. Khorramizadeh and N. Mahdavi-Amiri, "An efficient algorithm for sparse null space basis problem using abs methods," *Numerical Algorithms*, vol. 62, no. 3, pp. 469–485, 2013.

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

[25] Q. Peng and S. H. Low, "Distributed optimal power flow algorithm for radial networks, Ⅰ: Balanced single phase case," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 111–121, 2018.

[26] M. P. Abraham and A. A. Kulkarni, "ADMM-Based algorithm for solving DC-OPF in a large electricity network considering transmission losses," *IET Generation*, *Transmission & Distribution*, vol. 12, no. 21, pp. 5811–5823, 2018.

[27] Y. Zhang, M. Hong, E. Dall'Anese, S. V. Dhople, and Z. Xu, "Distributed controllers seeking ac optimal power flow solutions using ADMM," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4525–4537, 2018.

[28] 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.

[29] R. Hariharan, T. Kavitha, and K. Mehlhorn, "Faster algorithms for minimum cycle basis in directed graphs," *SIAM Journal of Computing*, vol. 38, no. 3, pp. 1430–1447, 2008.

[30] 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.

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

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. 3th ed, MIT, 2009.

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

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

[35] T. Leibfried, T. Mchedlidze, N. Meyer-Hübner, M. Nöllenburg, I. Rutter, P. Sanders, D. Wagner, and F. Wegner, "Operating power grids with few flow control buses," in *Proc. of the 6th ACM Int. Conf. on Future Energy Systems*, pp. 289–294, 2015.

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

Reza Asadi is with the Department of Computer Science at University of California, Irvine. He received the B.Sc. and M.Sc. degrees in computer science from Amirkabir University of Technology, Iran, in 2011 and 2013, respectively. His main research interests are in broad areas of distributed optimization and machine learning. His research focuses on developing scalable optimization and machine learning algorithms for large scale problems.

Solmaz S. Kia is with the Department of Mechanical and Aerospace Engineering, University of California, Irvine (UCI). She obtained the Ph.D. degree in mechanical and aerospace engineering from UCI, in 2009, and the M.Sc. and B.Sc. degrees in aerospace engineering from the Sharif University of Technology, Iran, in 2004 and 2001, respectively. She was a Senior Research Engineer at SySense Inc., El Segundo, CA from Jun. 2009-Sep. 2010. She held postdoctoral positions in the Department of Mechanical and Aerospace Engineering at the UC San Diego and UCI. She was the recipient of UC President's Postdoctoral Fellowship from 2012-2014. She is also a recipient of 2017 NSF CAREER award. Dr. Kia's main research interests, in a broad sense, include nonlinear control theory, distributed optimization/coordination/estimation, and probabilistic robotics.