

Multi-player minimum cost flow problems with nonconvex costs and integer flows

Shuvomoy Das Gupta & Lacro Pavel

Abstract—In this paper we consider a variant of the well known minimum cost flow problem in a directed network with nonconvex costs and integer flows. We formulate the problem in a multi-player setup, whereby we associate one player with each arc of the network. The goal of each player is to minimize its nonconvex cost that depends on the integer flow through the arc subject to the network flow constraints. In this multi-player setup, a Pareto optimal point is justified to be an efficient solution concept. We propose an algorithm to compute a Pareto optimal point. We show that, although the problem in its original form has coupled constraints binding every player, there exists an equivalent variable transformation that decouples the optimization problems for a number of players. Each of the decoupled players can solve its optimization problem in a decentralized manner. We use the solutions of those decoupled players to transform the optimization problems for the rest of the players using consensus constraints. Then we present algorithms based on algebraic geometry to find a Pareto optimal point.

I. INTRODUCTION

A *network flow problem* is a special class of optimization problems associated with some underlying directed network. Network flow problems have a remarkable range of applications [1]–[5]. The *minimum cost flow problem* is one of the most fundamental network flow problems [1]. It is associated with the flow of some commodity in a directed graph, where each arc of the network incurs a cost for the flow of that product, while satisfying the flow preservation laws in that network. The flow is often taken to be integer [1, Section 14.1], and can represent number of products, vehicles, data packets in communication networks *etc.* In many applications the cost function is nonconvex [6]–[10]. Nonconvex minimum cost flow problems have \mathcal{NP} -hard computational complexity [11]. Even when the cost function is convex, a polynomial time algorithm does not exist unless the cost function is of very specific structure [1].

This work was supported by NSERC-CRD and Thales, Inc. S. D. Gupta and L. Pavel are with the Edward S. Rogers Department of Electrical and Computer Engineering, University of Toronto shuvomoy.dasgupta@mail.utoronto.ca, pavel@control.utoronto.ca

In this paper we consider an extension of the minimum (nonconvex) cost flow problem with integer flows to a *multi-player setup* as follows. With each arc of the network graph we associate one player. Each of the players is trying to minimize its nonconvex cost function, subject to the network flow constraints. *Our goal is to seek a socially optimal solution in this multi-player problem.* A *vector optimal solution* that minimizes all the objectives simultaneously is unlikely to exist [12, page 176]. The celebrated *Nash equilibrium* [13] is also not very efficient, because the constraint set of the problem has equality constraints, thus making any feasible point a Nash equilibrium. Also, a Nash equilibrium does not necessarily correspond to a socially optimal outcome, as posteriori some of the players may decide to deviate from the Nash equilibrium in order to reduce their costs even more at the expense of the rest of the players' expense [14, Section 2.6.4]. In this paper, we rather consider seeking a *Pareto optimal point*. A Pareto optimal point is a socially optimal point which is feasible and where none of the players can improve their cost functions without strictly worsening some other player [12, Section 4.7.5]. So, in our search for socially optimal points we limit our search to Pareto optimal points.

Related work and contributions: In this paper, we propose an extension of the minimum (nonconvex) cost flow problem with integer flows in a multi-player setup and construct an algorithm to compute a Pareto optimal solution. Our problem can be interpreted as a multi-objective optimization problem [12, Section 4.7.5] with the objective vector consisting of a number of univariate nonconvex cost functions subject to the network flow constraints and integer flows. Finding Pareto optimal points in multi-objective network flow problems with integer flows has been limited to linear cost functions and two objectives [15]–[17]. However, our setup is single-commodity and each of the cost components is univariate due to the problem structure. We show that, although in its original form the problem has coupled constraints binding every player, there exists an equivalent variable transformation that decouples the optimization problems for a number of players. Solving

these decoupled optimization problems can potentially lead to a significant reduction in the number of candidate points to be searched. We use the solutions of these decoupled optimization problems to reduce the size of the set of candidate Pareto optimal solutions even further using algebraic geometry, and finally solve a sequence of univariate optimization problems for the rest of the players to find a Pareto optimal point. To the best of our knowledge, our methodology is novel.

The rest of the paper is organized as follows. Section II describes the problem. In Section III we show how to decouple the optimization problems for a number of players. Section IV transforms the optimization problems for the rest of the players using consensus constraints. In Section V we present the algorithms to compute a Pareto optimal point for our problem. Section VI presents an illustrative numerical example of our methodology in a transportation setup.

II. PROBLEM STATEMENT

A. Notation and notions

We denote the sets of real numbers, integers and natural numbers by \mathbf{R} , \mathbf{Z} and \mathbf{N} respectively. The set of consecutive integers from 1 to n is denoted by $[n] = \{1, 2, \dots, n\}$ and m to n is denoted by $[m : n] = \{m, m+1, \dots, n\}$. The i th column, j th row and (i, j) th component of a matrix $A \in \mathbf{R}^{m \times n}$ is denoted by A_j , a_j^T and a_{ij} . The submatrix of a matrix $A \in \mathbf{R}^{m \times n}$, which constitutes of its rows $r_1, r_1 + 1, \dots, r_2$, and columns $c_1, c_1 + 1, \dots, c_2$ is denoted by $A_{[r_1:r_2, c_1:c_2]}$. For $x, y \in \mathbf{R}^n$, $x \succeq y \Leftrightarrow (\forall i \in [n]) x_i \geq y_i$. Two copies of $x \in \mathbf{R}^n$ are denoted by $x^{(1)}$ and $x^{(2)}$.

B. Network structure

Let $G = (\mathcal{M}, \mathcal{A})$ be a directed connected graph associated with a network, where $\mathcal{M} = [m + 1]$ is the set of nodes, and $\mathcal{A} = [n]$ is the set of directed arcs. With each arc $j \in \mathcal{A}$, we associate one player (called the j th player). The variable controlled by the j th player is the nonnegative integer flow on arc j , denoted by $x_j \in \mathbf{Z}$. Each player minimizes a nonconvex cost function $f_j(x_j) : \mathbf{Z} \rightarrow \mathbf{R}$, subject to the network flow constraints. We assume each of the cost functions is proper, *i.e.*, for all $i \in [n]$ we have $-\infty \notin f_i(\mathbf{Z})$, and $\text{dom} f_i = \{x_i \in \mathbf{Z}^n \mid f_i(x_i) < +\infty\} \neq \emptyset$. There is an upper bound u_j , which limits how much flow the j th player can carry through arc j . Without any loss of generality we take the lower bound on every arc to be 0 [1, page 39]. The supply or demand of flow at each node $i \in \mathcal{M}$ is denoted by $b_i \in \mathbf{Z}$. If $b_i > 0$, then i is a supply node; if $b_i < 0$, then i is a demand node with a demand of $-b_i$, and if $b_i = 0$ then i is a transshipment node.

We allow parallel arcs to exist between two nodes. The vector formed by all the decision variables is denoted by $x = (x_1, x_2, \dots, x_n) \in \mathbf{Z}^n$. By $x_{-j} \in \mathbf{Z}^{n-1}$ we denote the vector formed by all the players' decision variables except x_j . To put emphasis on the j th player's variable we sometimes write x as (x_j, x_{-j}) .

The constraint set : The constraint set in any minimum cost flow problem consists of three types of constraints. They are:

(i) *Mass balance constraint.* This constraint states that for any node, the outflow minus inflow must equal the supply/demand of the node. We describe the constraint using node-arc incidence matrix. Let us fix a particular ordering of the arcs, and let $x \in \mathbf{Z}^n$ be the vector of flows that results, when the components x_j s are ordered accordingly. We define the *augmented node-arc incidence matrix* \tilde{A} with each row corresponding to a node and each column corresponding to an arc. The (i, j) th entry of \tilde{A} corresponds to i th node and j th arc, and is defined as follows.

$$\tilde{a}_{ij} = \begin{cases} 1, & \text{if } i \text{ is the start node of the } j\text{th arc,} \\ -1, & \text{if } i \text{ is the end node of the } j\text{th arc,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that parallel arcs will correspond to different columns with same entries in the matrix. So every column of \tilde{A} has exactly two nonzero entries, one equal to 1, one equal to -1 , indicating the start node and the end node of the associated arc. Denote, $\tilde{b} = (b_1, \dots, b_m, b_{m+1})$. Then, in matrix notation we write the mass balance constraint as $\tilde{A}x = \tilde{b}$. Note that the sum of the rows of \tilde{A} is equal to zero vector, so the rows are linearly dependent. By removing the last row of the linear system we can arrive at $Ax = b$, where, $A = \tilde{A}_{[1:m, 1:n]}$ and $b = \tilde{b}_{[1:m]}$. Now A , which is called the *node-arc incidence matrix*, is a full row rank matrix under the assumption of G being connected and $\sum_{i \in \mathcal{N}} b_i = 0$ [18, Corollary 7.1]. Note that, $\sum_{i \in \mathcal{N}} b_i = 0$, otherwise the problem is infeasible.

(ii) *Flow bound constraint.* The flow on any arc must be nonnegative and satisfy the capacity constraints, *i.e.*, $0 \preceq x \preceq u$.

(iii) *integrality constraint.* The flow on any arc is integer, *i.e.*, $x \in \mathbf{Z}^n$.

C. Multi-player problem

The goal of the i th player for $i \in [n]$, given other players' strategies $x_{-i} \in \mathbf{Z}^{n-1}$, is to solve the minimization

problem

$$\begin{aligned}
& \text{minimize}_{x_i} && f_i(x_i) \\
& \text{subject to} && Ax = A_i x_i + \sum_{j=1, j \neq i}^n A_j x_j = b \\
& && 0 \preceq (x_i, x_{-i}) \preceq u \\
& && (x_i, x_{-i}) \in \mathbf{Z}^n.
\end{aligned} \tag{1}$$

So the constraint set, which we denote by P can be written as, $P = \{x \in \mathbf{Z}^n \mid Ax = b, 0 \preceq x \preceq u\}$ and the subset of P containing only the equality constraints is denoted by $Q = \{x \in \mathbf{Z}^n \mid Ax = b\}$. Our objective is to devise an algorithm to calculate Pareto optimal point(s) for (1). A Pareto optimal point is defined as follows.

Definition 1: In (1) a point $x^* \in P$ is called Pareto optimal if it satisfies the following: there *does not* exist another point $\tilde{x} \in P$ such that $(\forall i \in [n]) \quad f_i(\tilde{x}_i) \leq f_i(x_i^*)$, with at least one $j \in [n]$ satisfying $f_j(\tilde{x}_j) < f_j(x_j^*)$.

III. DECOUPLING THE LAST $n - m$ OPTIMIZATION PROBLEMS

In this section, we show how to decouple the optimization problems for the last $n - m$ players. First, we present the following lemma.

Lemma 1: Consider a node-arc incidence matrix A . Then from A , we can extract a unimodular $m \times m$ square submatrix B .

Proof: We omit the proof due to space constraints. ■

Without any loss of generality, we rearrange and reindex the columns of the matrix A so that $A = [B \mid A_{m+1} \mid \dots \mid A_n]$. Now we have the following lemma.

Lemma 2: Let $C = B^{-1}A$ and $d = B^{-1}b$. Then, $C \in \mathbf{Z}^{m \times n}$, $d \in \mathbf{Z}^m$, and the constraint set P and Q has the equivalent representation: $P = \{x \in \mathbf{Z}^n \mid Cx = d, 0 \preceq x \preceq u\}$ and $Q = \{x \in \mathbf{Z}^n \mid Cx = d\}$.

Proof: We omit the proof due to space constraints. ■

Lemma 3: Consider the unimodular matrix

$$U = \begin{bmatrix} I_{m \times m} & -B^{-1}A_{[1:m, m+1:n]} \\ 0_{n-m \times m} & I_{n-m \times n-m} \end{bmatrix}.$$

Then, $CU = [I_{m \times m} \mid 0_{m \times n-m}]$.

Proof: Follows from multiplying $C = [I_{m \times m} \mid B^{-1}A_{m+1} \mid \dots \mid B^{-1}A_n]$ with U . ■

The following theorem is key in transforming problem (1) into an equivalent form with $n - m$ decoupled optimization problems for players $i \in [m + 1 : n]$.

Theorem 1: The constraint set Q is nonempty and for any x , $x \in Q$ there exists $z \in \mathbf{Z}^{n-m}$ such that

$$x = (d_1 - h_1^T z, \dots, d_m - h_m^T z, z_1, \dots, z_{n-m}), \tag{2}$$

where d_i is the i th component of $d = B^{-1}b$, and $h_i^T \in \mathbf{Z}^{n-m}$ is the i th row of $B^{-1}A_{[1:m, m+1:n]}$.

Proof: We omit the proof due to space constraints. ■

Remark: From Theorem 1, $x_i = z_{i-m}$ for $i \in [m + 1 : n]$. So, in (1) we can write the optimization problems for any player $m + i$ for $i \in [1 : n - m]$ in the new variable z as follows.

$$\begin{aligned}
& \text{minimize}_{z_i} && f_i(z_i) \\
& \text{subject to} && 0 \leq z_i \leq u_i \\
& && z_i \in \mathbf{Z}.
\end{aligned} \tag{3}$$

Each of these $n - m$ optimization problems is a decoupled univariate optimization problem, which we can easily solve graphically or by sorting. Doing so immediately reduces the feasible set into a much smaller set. Let us denote the sorted set of different optimal solutions for player $m + i$ for $i \in [n - m]$ as $D_i = \{z_{i,1}, z_{i,2}, \dots, z_{i,p_i}\}$, where p_i is the total number of minimizers. Define, $D = \times_{i=1}^{n-m} D_i$, where \times denotes the Cartesian product.

IV. CONSENSUS REFORMULATION FOR THE FIRST m PLAYERS

In this section we transform the optimization problems for the first m players in z using consensus constraints. Consider the optimization problems for the first m players in variable z , which have coupled costs due to (2). We deal with the issue by introducing *consensus constraints* [19, Section 5.2]. We provide each player $i \in [m]$ with its own local copy of z , denoted by $z^{(i)} \in \mathbf{Z}^{n-m}$, which acts as its decision variable. This local copy has to satisfy the following conditions. *First*, using (2) for any $i \in [m]$, $x_i = d_i - h_i^T z^{(i)}$. The copy $z^{(i)}$ has to be in consensus with the rest of the first m players, i.e., $z^{(i)} = z^{(j)}$ for all $j \in [m] \setminus \{i\}$. *Second*, the copy $z^{(i)}$ has to satisfy the flow bound constraints, i.e., $0 \leq d_i - h_i^T z^{(i)} \leq u_i$ for all $i \in [m]$. *Third*, for the last $n - m$ players $z_i \in D_i$, as obtained from the solutions of the decoupled optimization problems (3), so $z^{(i)}$ has to be in D , i.e., $z^{(i)} = z \in D$ for all $i \in [m]$. So, for all $i \in [m]$, the i th player's optimization problem in variable $z^{(i)}$ can be written as:

$$\begin{aligned}
& \text{minimize}_{z^{(i)}} \quad \bar{f}_i(z^{(i)}) = f_i(d_i - h_i^T z^{(i)}) \\
& \text{subject to} \quad z^{(i)} = z^{(j)}, \quad j \in [m] \setminus \{i\} \\
& \quad \quad \quad 0 \leq d_i - h_i^T z^{(i)} \leq u_i \\
& \quad \quad \quad z_j^{(i)} \in D_j, \quad j \in [n-m].
\end{aligned} \tag{4}$$

We can write the last two constraints in (4) in polynomial forms as $q_i(z^{(i)}) = \prod_{l=0}^{u_i} (d_i - h_i^T z^{(i)} - l)$ and $r_j(z^{(i)}) = \prod_{l=1}^{p_i} (z_j^{(i)} - z_{j,l})$ for $j \in [n-m]$. Hence, for all $i \in [m]$ any feasible $z^{(i)}$ for problem (4) comes from the following set:

$$\begin{aligned}
\mathcal{F} = \{z \in \mathbf{Z}^{n-m} \mid (\forall i \in [m]) \quad & q_i(z) = 0, \\
& (\forall j \in [n-m]) \quad r_j(z) = 0\}
\end{aligned} \tag{5}$$

In (5), the intersection in the first line ensures that the consensus constraints are satisfied, and the second line just expands the first. So, the optimization problem (4) is equivalent to

$$\begin{aligned}
& \text{minimize}_{z^{(i)}} \quad \bar{f}_i(z^{(i)}) \\
& \text{subject to} \quad z^{(i)} \in \mathcal{F},
\end{aligned} \tag{6}$$

for $i \in [m]$, i.e., each of these players are optimizing over a *common constraint set* \mathcal{F} . So, finding the points in \mathcal{F} is of interest.

V. ALGORITHMS TO COMPUTE A PARETO OPTIMAL POINT

This section is organized as follows. First, we review some necessary background on algebraic geometry. Then we present a theorem to check if \mathcal{F} is nonempty, and provide an algorithm to compute the points in a nonempty \mathcal{F} . Finally, we present our algorithm to compute Pareto optimal point(s) for our problem.

A. Background on algebraic geometry

A *monomial* in variables $x = (x_1, x_2, \dots, x_n)$ is a product of the structure $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$, where $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbf{N}^n$. A *polynomial* is a linear combination of monomials. The set of all real polynomials in $x = (x_1, \dots, x_n)$ with complex coefficients is denoted by $\mathbf{C}[x]$ with the variable ordering $x_1 > x_2 > \dots > x_n$. The ideal generated by $f_1, \dots, f_m \in \mathbf{C}[x]$ is the set $\mathbf{ideal}\{f_1, \dots, f_m\} = \{\sum_{i=1}^m h_i f_i \mid (\forall i \in [m]) \quad h_i \in \mathbf{C}[x]\}$, and the *affine variety* V of f_1, \dots, f_m is the set

$$V(f_1, \dots, f_m) = \{x \in \mathbf{C}^n \mid (\forall i \in [m]) \quad f_i(x) = 0\}. \tag{7}$$

A *monomial order* on $\mathbf{C}[x_1, \dots, x_n]$ is a relation, denoted by \succ , on the set of monomials $x^\alpha, \alpha \in \mathbf{N}^n$ satisfying the following: (i) it is a total order, (ii) every

nonempty subset of \mathbf{N}^n has a smallest element under \succ and (iii) $x^\alpha \succ x^\beta \Rightarrow x^{\alpha+\gamma} \succ x^{\beta+\gamma}$, where x^γ is any monomial. We will use *lexicographic order*, where we say $x^\alpha \succ_{\text{lex}} x^\beta$ if and only if the left most nonzero entry of $\alpha - \beta$ is positive. Suppose we are given a monomial order \succ and a polynomial $f(x) = \sum_{\alpha \in S} f_\alpha x^\alpha$. The *leading term* of the polynomial with respect to \succ , denoted by $\text{lt}_\succ(f)$, is that monomial $f_\alpha x^\alpha$ with $f_\alpha \neq 0$, such that $x^\alpha \succ x^\beta$ for all other monomials x^β with $f_\beta \neq 0$. The monomial x^α is called the *leading monomial* of f . Consider a nonzero ideal $I = \mathbf{ideal}\{f_1, \dots, f_m\}$. The set of the leading terms for the polynomials in I is denoted by $\text{lt}_\succ(I)$. Thus $\text{lt}_\succ(I) = \{cx^\alpha \mid (\exists f \in I) \text{lt}_\succ(f) = cx^\alpha\}$. By $\mathbf{ideal}\{\text{lt}_\succ(I)\}$ with respect to \succ we denote the ideal generated by the elements of $\text{lt}_\succ(I)$. A *Groebner basis* G_\succ of an ideal I with respect to monomial order \succ is a finite set of polynomials $f_1, \dots, f_m \in I$ such that $\mathbf{ideal}\{\text{lt}_\succ(I)\} = \mathbf{ideal}\{\text{lt}_\succ(f_1), \dots, \text{lt}_\succ(f_m)\}$. A *reduced Groebner basis* $G_{\text{reduced}, \succ}$ is a Groebner basis with respect to monomial order \succ such that, for any $f \in G_{\text{reduced}, \succ}$ the coefficient associated with $\text{lt}_\succ(f)$ is 1, and for all $f \in G_{\text{reduced}, \succ}$ no monomial of f lies in $\mathbf{ideal}\{\text{lt}_\succ(G \setminus \{f\})\}$. For a nonzero ideal I and given monomial ordering the reduced Groebner basis is unique [20, Page 92, Proposition 6]. A Groebner basis with respect to lexicographic order is denoted by $G_{\text{reduced}, \succ_{\text{lex}}}$. For any $l \in [0 : n-1]$, $I_l = I \cap \mathbf{C}[x_{l+1}, \dots, x_n]$ is the l th *elimination ideal* of the ideal I , and $G_l = G_{\text{reduced}, \succ_{\text{lex}}} \cap \mathbf{C}[x_{l+1}, \dots, x_n]$ is a Groebner basis for I_l .

B. Nonemptiness of \mathcal{F}

Using the definition of affine variety in (7), we can write \mathcal{F} in (5) as follows.

$$\mathcal{F} = V(q_1, \dots, q_m, r_1, \dots, r_{n-m}) \cap \mathbf{Z}^{n-m}. \tag{8}$$

So finding points in \mathcal{F} is equivalent to finding integer points in a certain affine variety. Whether \mathcal{F} is nonempty can be checked by the following theorem.

Theorem 2: The set \mathcal{F} is nonempty if and only if

$$G_{\text{reduced}, \succ} \neq \{1\},$$

where $G_{\text{reduced}, \succ}$ is the reduced Groebner basis of $\mathbf{ideal}\{q_1, \dots, q_m, r_1, \dots, r_{n-m}\}$ with respect to any ordering.

Proof: We omit the proof here because of space limitation. ■

Lemma 4: Suppose $G_{\text{reduced}, \succ} \neq \{1\}$. Then $\mathcal{F} = V(G_{\text{reduced}, \succ_{\text{lex}}})$.

Proof: We omit the proof due to space limitation. ■

Now we present Algorithm 1 that can calculate all the points in a nonempty \mathcal{F} .

Algorithm 1 Extracting the points in \mathcal{F}

Input: Polynomial system $q_i(z) = 0$ for $i \in [m]$ and $r_j(z) = 0$ for $j \in [n - m]$, $G_{\text{reduced}, \succ_{\text{lex}}} \neq \{1\}$.

Output: The set \mathcal{F} .

Algorithm:

- Calculate the set

$$G_{n-m-1} = G_{\text{reduced}, \succ_{\text{lex}}} \cap \mathbf{C}[z_{n-m}],$$

which is a Groebner basis of the $(n - m)$ th elimination ideal of **ideal** $\{q_1, \dots, q_m, r_1, \dots, r_{n-m}\}$ and consists of univariate polynomials in z_{n-m} as an implication of [20, page 116, Theorem 2]. Find the variety of G_{n-m-1} , denoted by $V(G_{n-m-1})$, which will contain the list all possible z_{n-m} coordinates for the points in \mathcal{F} .

- Calculate

$$G_{n-m-2} = G_{\text{reduced}, \succ_{\text{lex}}} \cap \mathbf{C}[z_{n-m-1}, z_{n-m}],$$

which is again a Groebner basis of the $(n - m - 1)$ th elimination ideal of

$$\mathbf{ideal} \{q_1, \dots, q_m, r_1, \dots, r_{n-m}\}$$

and consists of bivariate polynomials in z_{n-m} and z_{n-m-1} . From Step 1, we already have the z_{n-m} coordinates for the points in \mathcal{F} . So, by substituting those $|V(G_{n-m-1})|$ values in G_{n-m-2} , denoted by we arrive at set of a set of univariate polynomials in z_{n-m-1} , denoted by $\{\bar{G}_{n-m-2}^{(i)}\}_{i=1}^{|V(G_{n-m-1})|}$. For all $i = 1, 2, \dots, |V(G_{n-m-1})|$, find the variety of $\bar{G}_{n-m-2}^{(i)}$, denoted by $V(\bar{G}_{n-m-2}^{(i)})$ which will contain the list all possible z_{n-m-1} coordinates associated with a particular $z_{n-m} \in V(G_{n-m-1})$. Now we have all the possible (z_{n-m-1}, z_{n-m}) coordinates.

- We repeat this procedure for $G_{n-m-3}, G_{n-m-4}, \dots, G_0$. In the end, we have all the points in \mathcal{F} .

return \mathcal{F} .

Lemma 5: Algorithm 1 correctly calculates all the points in \mathcal{F} when it is nonempty.

Proof: We omit the proof here because of space limitation. ■

C. Finding the Pareto optimal points from \mathcal{F}

Suppose, $G_{\text{reduced}, \succ_{\text{lex}}} \neq \{1\}$, and using Algorithm 1 we have computed the elements of \mathcal{F} . Now we propose Algorithm 2 and show that the resultant points are Pareto optimal.

Lemma 6: Consider (9)-(12) in Algorithm 2. For all $i \in [m - 1]$ we have $\mathcal{F}_{s_{i+1}}^* \subseteq \mathcal{F}_{s_i}^* \subseteq \mathcal{F}$.

Proof: Follows from (11), (9) and (12). ■

Lemma 7: Consider (9)-(12) in Algorithm 2. For any $i \in [m]$, $x_{s_i} \in X_{s_i}^*$ if and only if $z^* \in \mathcal{F}_{s_i}^*$. Furthermore, $z^* \in \mathcal{F}_{s_i}^*$ solves the following optimization problem

$$\begin{aligned} & \text{minimize}_z \quad f_{s_i}(d_{s_i} - h_{s_i}^T z) \\ & \text{subject to} \quad z \in \mathcal{F}_{s_{i-1}}^*, \end{aligned}$$

Algorithm 2 Computing the set of solutions to problem (6).

Input: The optimization problem (6) for any $i \in [m]$, $\mathcal{F} \neq \emptyset$.

Output: Pareto optimal solutions for problem (1).

Algorithm:

for $i \in [m]$

$$\begin{aligned} X_i & := \{d_i - h_i^T z^{(i)} \mid z^{(i)} \in \mathcal{F}\}, \\ (\forall x_i \in X_i) \quad (X_i)^{-1}(x_i) & := \{z^{(i)} \in \mathcal{F} \mid x_i = d_i - h_i^T z^{(i)}\}. \end{aligned} \quad (9)$$

end for

Sort the elements of the $\{X_i\}_{i=1}^m$ s with respect to cardinality of the elements in a descending order. Denote the index set of the sorted set by $\{s_1, s_2, \dots, s_m\}$ such that it is a partition of $[m]$

$$|X|_{s_1} \geq |X|_{s_2} \geq \dots \geq |X|_{s_m}.$$

for $i \in [m]$

Solve the univariate optimization problem

$$\begin{aligned} & \text{minimize}_{x_{s_i}} \quad f_{s_i}(x_{s_i}) \\ & \text{subject to} \quad x_{s_i} \in X_{s_i}, \end{aligned} \quad (10)$$

and denote the set of solutions by $X_{s_i}^*$.

$$\mathcal{F}_{s_i}^* := \bigcup_{x_{s_i} \in X_{s_i}^*} (X_{s_i}^*)^{-1}(x_{s_i}) \subseteq \mathcal{F}, \quad (11)$$

$$X_{s_{i+1}} := \{d_{s_{i+1}} - h_{s_{i+1}}^T z \mid z \in \mathcal{F}_{s_i}^*\} \quad (12)$$

end for

return $\mathcal{F}_{s_m}^*$.

for all $i = 2, \dots, m$.

Proof: We omit the proof here due to space constraints. ■

Note that in Algorithm 2, at every stage $\mathcal{F}_{s_i}^*$ stays nonempty for any $i \in [m]$ as shown by the following lemma.

Lemma 8: Suppose $\mathcal{F} \neq \emptyset$. Then in Algorithm 2 $\mathcal{F}_{s_i}^*$ is nonempty for any $i \in [m]$.

Proof: We omit the proof due to space limitation. ■

Theorem 3: For any $z^* \in \mathcal{F}_{s_m}^*$, $x^* = (d_1 - h_1^T z^*, \dots, d_m - h_m^T z^*, z_1^*, \dots, z_{n-m}^*)$ is a Pareto optimal point.

Proof: We omit the proof due to space constraints. ■

VI. NUMERICAL EXAMPLE FOR A TRANSPORTATION PROBLEM

In this section we present an illustrative numerical example of our methodology in a transportation setup. Figure 1 shows a randomly generated directed network

Cost function (i th row for player i)	
$-\frac{x_1^4}{30} - \frac{13x_1^3}{15} + \frac{259x_1^2}{30} - \frac{263x_1}{15} + 1$	
$\frac{77x_2^5}{120} - \frac{247x_2^4}{24} + \frac{471x_2^3}{8} - \frac{3365x_2^2}{24} + \frac{6779x_2}{60} + 1$	
$\frac{47x_3^4}{24} - \frac{133x_3^3}{4} + \frac{4897x_3^2}{24} - \frac{2123x_3}{4} + 485$	
$\frac{323x_4^5}{3360} - \frac{2179x_4^4}{1120} + \frac{47393x_4^3}{3360} - \frac{48709x_4^2}{1120} + \frac{7885x_4}{168} + 5$	
$(x_5 - 1)^2$	
$-\frac{x_6^4}{8} + \frac{25x_6^3}{12} - \frac{71x_6^2}{8} + \frac{95x_6}{12} + 10$	
$ x_7 - 5 $	
$\frac{11x_8^7}{1260} - \frac{7x_8^6}{36} + \frac{119x_8^5}{72} - \frac{479x_8^4}{72} + \frac{4609x_8^3}{360} - \frac{803x_8^2}{72} + \frac{155x_8}{28} + 1$	
$-\frac{15x_9^3}{16} + \frac{365x_9^2}{16} - \frac{2865x_9}{16} + \frac{7315}{16}$	
$(x_{10} - 10)^2$	
$\frac{5x_{11}^4}{6} - \frac{35x_{11}^3}{3} + \frac{355x_{11}^2}{6} - \frac{370x_{11}}{3} + 90$	
$\frac{5x_{12}^4}{6} - \frac{25x_{12}^3}{3} + \frac{175x_{12}^2}{6} - \frac{110x_{12}}{3} + 15$	
$\frac{5x_{13}^4}{6} - 15x_{13}^3 + \frac{595x_{13}^2}{6} - 280x_{13} + 285$	
$\frac{5x_{14}^4}{6} - \frac{85x_{14}^3}{3} + \frac{2155x_{14}^2}{6} - \frac{6020x_{14}}{3} + 4165$	
$ x_{15} - 7 $	
$\begin{cases} x_{16} + 1, & \text{if } 0 \leq x_{16} \leq 3 \\ 0, & \text{if } 4 \leq x_{16} \leq 6 \\ (x_{16} + 1)^3, & \text{if } 7 \leq x_{16} \leq 9 \\ -\frac{x_{16}^3}{6} + \frac{13x_{16}^2}{2} - \frac{244x_{16}}{3} + 330, & \text{else} \end{cases}$	

TABLE I: Cost functions for the numerical example

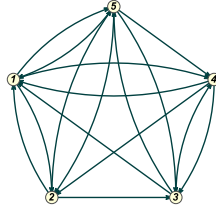


Fig. 1: Network in consideration

with 5 nodes and 16 arcs. Nodes 2 and 4 represent two retail centers with demands for 13 and 11 units of a certain product. The warehouses are denoted by nodes 1 and 3, which supply 9 and 15 units respectively. Node 5 is a transshipment node. Different modes of shipment from one node to other is represented by the arcs in the figure, and each of these shipments are carried out by different organizations. The cost of a certain shipment depends on the number of products shipped and is nonlinear and not necessarily convex. With each arc we associate one player. Each of the players is trying to minimize its cost. We seek a Pareto optimal point in this setup. We have used Wolfram Mathematica 10 for numerical calculation. The resource vector is manually constructed (to have a feasible system) and is given by $(9, -13, 15, -11)$. The upper bound for the decision vector is randomly generated and is given by

$(5, 6, 6, 10, 10, 7, 11, 13, 16, 12, 4, 5, 6, 14, 13, 15)$. The cost functions for the players are listed in Table I. All of the cost functions are randomly generated except for players 5,7,10,15 and 16. Now we apply Algorithms 1 and 2 to find the Pareto optimal points, which are $(1,3,5,4,11,10,2,1,3,7,7,5)$ and $(1,3,5,6,11,10,2,1,3,7,7,6)$.

REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [2] S. Sojoudi and J. Lavaei, "Convexification of generalized network flow problem with application to power systems," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 7552–7559.
- [3] E. Wei and C. Bandi, "Fairness considerations in network flow problems," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, Dec 2015, pp. 6909–6914.
- [4] A. K. Bangla and D. Castanon, "Accelerated dual coordinate algorithms for separable convex cost network flow problems," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4468–4473.
- [5] R. Zhang, K. Spieser, E. Frazzoli, and M. Pavone, "Models, algorithms, and evaluation for autonomous mobility-on-demand systems," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 2573–2587.
- [6] T. L. Magnanti and R. T. Wong, "Network design and transportation planning: Models and algorithms," *Transportation science*, vol. 18, no. 1, pp. 1–55, 1984.
- [7] S. C. Graves and J. B. Orlin, "A minimum concave-cost dynamic network flow problem with an application to lot-sizing," *Networks*, vol. 15, no. 1, pp. 59–71, 1985.
- [8] R. M. Soland, "Optimal facility location with concave costs," *Operations Research*, vol. 22, no. 2, pp. 373–382, 1974.
- [9] M. S. Daskin, *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 2011.
- [10] W. I. Zangwill, "Minimum concave cost flows in certain networks," *Management Science*, vol. 14, no. 7, pp. 429–450, 1968.
- [11] M. R. Garey and D. S. Johnson, "Computers and intractability: a guide to the theory of \mathcal{NP} -completeness. 1979," *San Francisco, CA: Freeman*, 1979.
- [12] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [13] J. F. Nash, "Equilibrium points in n-person games," *Proc. Nat. Acad. Sci. USA*, vol. 36, no. 1, pp. 48–49, 1950.
- [14] L. Pavel, *Game theory for control of optical networks*. Springer Science & Business Media, 2012.
- [15] A. Raith and M. Ehrgott, "A two-phase algorithm for the biobjective integer minimum cost flow problem," *Computers & Operations Research*, vol. 36, no. 6, pp. 1945–1954, 2009.
- [16] H. Lee and P. S. Pulat, "Bicriteria network flow problems: Integer case," *European Journal of Operational Research*, vol. 66, no. 1, pp. 148–157, 1993.
- [17] A. Eusébio and J. R. Figueira, "Finding non-dominated solutions in bi-objective integer network flow problems," *Computers & Operations Research*, vol. 36, no. 9, pp. 2554–2564, 2009.
- [18] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.
- [19] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [20] D. Cox, J. Little, and D. O'shea, *Ideals, varieties, and algorithms*. Springer, 1992, vol. 3.