Optim Lett (2012) 6:1173–1184
DOI 10.1007/s11590-011-0356-5

ORIGINAL PAPER

A network simplex based algorithm for the minimum cost proportional flow problem with disconnected subnetworks

Ufuk Bahçeci · Orhan Feyzioğlu

Received: 23 September 2010 / Accepted: 1 June 2011 / Published online: 17 June 2011

Abstract In this study, we present a variant of the minimum cost network flow problem where the associated graph contains several disconnected subgraphs and it is required that the flows on arcs belonging to same arc subsets to be proportional. This type of network is mostly observed in large supply chains of assemble-to-order products. It is shown that any feasible solution of a reformulation of this problem has a special characteristic. By taking into account this fact, a network simplex based primal simplex algorithm is developed and its details are provided.

Keywords Minimum cost network flow · Proportional flow · Disconnected subnetworks · Network simplex algorithm

1 Introduction

In contemporary supply networks, a product is generally manufactured from several subparts, each supplied from separate networks. However if the bill-of-materials structure is taken into account, these subnetworks should be integrated so as to provide parts in *proportional* amounts to build a single product. Otherwise, unnecessary component stocks are held and transferred, or final products could not be build due to the shortage of components [4,7]. Motivated by those practical points, a new mathematical programming model is defined within a general framework and a solution algorithm is proposed in this study.

U. Bahçeci · O. Feyzioğlu (✉)
Department of Industrial Engineering, Galatasaray University,
Çırağan Cad. No:36 Ortaköy, 34357 İstanbul, Turkey
e-mail: ofeyzioglu@gsu.edu.tr

U. Bahçeci
e-mail: ubahceci@gsu.edu.tr

Let $\mathcal{G}(\mathcal{N}, \mathcal{A})$ be a directed network where $\mathcal{N} = \{1, \ldots, n\}$ is the set of nodes and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}\}$ is the set of directed arcs. Let $\mathcal{G}_k(\mathcal{N}_k, \mathcal{A}_k)$ be a subnetwork of $\mathcal{G}$ where $\mathcal{N}_k = \{1, \ldots, n_k\}$ is the set of nodes and $\mathcal{A}_k = \{(i, j) : i, j \in \mathcal{N}_k\}$ is the set of arcs corresponding to subnetwork $k \in \mathcal{K} = \{1, \ldots, m\}$ with the property that $\mathcal{G}_k \cap \mathcal{G}_{k'} = \emptyset$ for each $k \neq k'$ and $k, k' \in \mathcal{K}$, and $\cup_{k\in\mathcal{K}}\mathcal{G}_k = \mathcal{G}$. We define u_{ij}, c_{ij} and p_{ij} as being the flow upper bound, unit flow cost and proportionality coefficient related to arc $(i, j) \in \mathcal{A}$ respectively. Let $b_i > 0$ if node $i \in \mathcal{N}$ is a supply node, and $b_i < 0$ if node $i \in \mathcal{N}$ is a demand node. It is assumed that $\sum_{i\in\mathcal{N}_k} b_i = 0$ for all $k \in \mathcal{K}$. Let $\mathcal{A}'_{sk}$ be the subset of arcs in $\mathcal{A}_k$ and $\mathcal{A}'_s = \cup_k \mathcal{A}'_{sk}$ be the set of all arcs in $\mathcal{A}$ that should have proportional flow for requirement s respectively where $s \in \mathcal{S} = \{1, \ldots, t\}$. In problem P1, which we call the *minimum cost proportional flow problem with disconnected subnetworks* (MCPFD), we want to find the amount of flows f_{ij} on each arc $(i, j) \in \mathcal{A}$ so that

$$\text{P1: } \min \sum_{(i,j)\in\mathcal{A}} c_{ij} f_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j:(i,j)\in\mathcal{A}_k} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}_k} f_{ji} = b_i \quad i \in \mathcal{N}_k, k \in \mathcal{K}, \tag{2}$$

$$f_{ij}/p_{ij} \text{ are all equal} \qquad (i, j) \in \mathcal{A}'_s, s \in \mathcal{S}, \tag{3}$$

$$0 \leqslant f_{ij} \leqslant u_{ij} \qquad (i, j) \in \mathcal{A}_k, k \in \mathcal{K}. \tag{4}$$

In this problem, the objective is to minimize the total cost associated with the flow on arcs while constraints (2) are for the flow conservation, constraints (3) are the proportional flow constraints, and constraints (4) impose simple lower and upper bounds on the arc flows. If the constraints (3) were not included in problem P1, then we would get the well-known minimum cost network flow problem (MCNF). There exist efficient algorithms to deal with MCNF such as network simplex algorithm [1,6]. However, the addition of proportionality constraints complicates the problem and the available solution procedures developed for MCNF can not be applied directly. When the number of nodes is at least an order of magnitude larger than the number of side constraints, it becomes possible to exploit the special network structure [9].

Several variants of the MCNF problem with side constraints have been studied. Based on the relaxation and decomposition techniques, Ali et al. [3] solved the equal flow problem in which selected pairs of arcs are required to have identical flow. Ahuja et al. [2] introduced the simple equal flow problem in which only a single set of arcs is required to have identical flow and developed special purpose primal simplex algorithm. Calvete [5] introduced the general equal flow problem which extends the simple equal flow problem by allowing multiple sets of arcs to have identical flow. This former problem corresponds to a special case of the problem P1.

Along this line of research, the manufacturing network flow (MNF) model is introduced by Fang and Qi [8] where synthesis of different materials to a single product and the distilling of one material to many different products can be realized. The authors modified the network simplex method according to this special flow problem and solved a simplified version of their model. Mo et al. [13] considered an integrated manufacturing supply chain where multiple products are manufactured across multiple

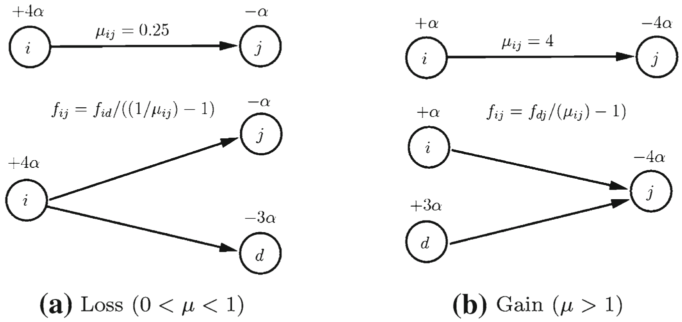


Fig. 1 Modelling generalized networks with linear gain/loss

manufacturing plants by distilling a unique raw material, and similar to [2,5], they presented a modified network simplex method which exploits the special structure of basis.

Later, Mo et al. [14] expanded the MNF model by incorporating certain features of the ordinary multi-commodity network flow models. Lu et al. [11] studied a manufacturing network flow model in which the assumption requiring the total flow in and out of a node to be equal (mass balance constraint) is relaxed. Venkateshan et al. [16] developed a network-simplex-based algorithm based on efficient data structures to solve a minimum cost flow problem formulated on such generalized networks. More recently, Lu et al. [12] proposed an algorithmic method to obtain an initial basic feasible solution to start the existing network simplex algorithm, and also presented a network based approach to check the dual feasibility conditions. Wang and Lin [17] proposed a network simplex algorithm with detailed graphical operations for solving the minimum distribution cost problem which is indeed a specialized MNF problem containing both distillation and common (source, sink and transhipment) nodes.

It is possible to transform an instance of the MNF problem to an instance of the MCPFD problem, and vice versa. The necessary transformation steps are given in the Appendix. However, when an instance of the MCPFD problem is transformed, the methods proposed in the literature to solve the MNF problem will be not more efficient than the method introduced in this study and that is specially designed to solve the MCPFD problem itself. This is because when the number of disconnected subnetworks will be high, being able to consider the disconnected network topology explicitly in the solution algorithm becomes more attractive in terms of execution times.

Note that the generalized network models with linear gains and/or losses can be also transformed into MCPFD as illustrated in Fig. 1. In that figure, the amount indicated over a node corresponds to its supply or demand, μ_{ij} is the gain/loss factor associated with arc (i, j), and node d is a dummy node.

The flow on all arcs in a proportional flow subset can also be viewed as a single decision variable. Let $c_s = \sum_{(i,j)\in\mathcal{A}'_s} p_{ij}c_{ij}$ and $u_s = \min_{(i,j)\in\mathcal{A}'_s} u_{ij}/p_{ij}$ for all $s \in \mathcal{S}$. Let $\mathbf{a}_{ij}$ denotes the column associated to arc (i, j) in the node-arc incidence matrix of network $\mathscr{G}$ and a^l_{ij} be the l-th component of vector $\mathbf{a}_{ij}$. Let also $\mathbf{a}_s = \sum_{(i,j)\in\mathcal{A}'_s} p_{ij}\mathbf{a}_{ij}$ for all $s \in \mathcal{S}$ and a^l_s be the l-th component of vector $\mathbf{a}_s$. As $a^i_{ij} = 1$, $a^j_{ij} = -1$, $a^l_{ij} = 0$ for each $l \neq i, j \in \mathcal{N}_k$ and $(i, j) \in \mathcal{A}'_{sk}$ and all the subnetworks are disconnected, we observe for each subnetwork k that

$$\sum_{l \in \mathcal{N}_k} a_s^l = \sum_{(i,j) \in \mathcal{A}'_{sk}} \sum_{l \in \mathcal{N}_k} a_{ij}^l = 0 \qquad s \in \mathcal{S}. \tag{5}$$

Let $\tilde{\mathcal{G}}(\mathcal{N}, \tilde{\mathcal{A}})$ be the network with $\tilde{\mathcal{A}} = \mathcal{A} \backslash \cup_{s \in \mathcal{S}} \mathcal{A}'_s$ and $\tilde{\mathcal{G}}_k(\mathcal{N}_k, \tilde{\mathcal{A}}_k)$ be the network with $\tilde{\mathcal{A}}_k = \mathcal{A}_k \backslash \cup_{s=1}^t \mathcal{A}'_{sk}$ for all $k \in \mathcal{K}$. Then, problem P1 can be transformed into the problem P2 as follows:

$$\text{P2: } \min \sum_{(i,j) \in \tilde{\mathcal{A}}} c_{ij} f_{ij} + \sum_{s=1}^{t} c_s f_s \tag{6}$$

$$\text{s.t.} \sum_{j:(i,j) \in \tilde{\mathcal{A}}} f_{ij} - \sum_{j:(j,i) \in \tilde{\mathcal{A}}} f_{ji} + \sum_{s=1}^{t} a_s^i f_s = b_i \quad i \in \mathcal{N}_k, k \in \mathcal{K} \tag{7}$$

$$0 \leqslant f_{ij} \leqslant u_{ij} \qquad (i,j) \in \tilde{\mathcal{A}}_k, k \in \mathcal{K} \tag{8}$$

$$0 \leqslant f_s \leqslant u_s \qquad s \in \mathcal{S} \tag{9}$$

After this reformulation, we need to consider only problem P2 for further analysis. Without loss of generality, it is assumed that each arc can appear in at most one proportional flow subset. Otherwise, if any arc shows up in multiple proportional flow subsets, then the flows on all arcs in these subsets will be proportional. This further implies that all of these arcs could be regrouped into one subset, and our assumption will be again satisfied.

2 Structure of the basis

It must be noted that if $|\mathcal{S}| = t \geqslant n = |\mathcal{N}|$, then there will be many basic feasible solutions of P2 in (6)–(9) involving only variables $\{f_1, \ldots, f_t\}$, and the simplex algorithm will pivot between these basic feasible solutions. In turn, this will reduce the efficiency of the proposed algorithm since there will be no possibility to take the advantage of network structure. Therefore, we assume that $t < n$ holds.

Lemma 1 *The rank of the matrix* $\mathbf{A}$ *corresponding to constraints (7) is equal to* $n - m$.

Proof The matrix $\mathbf{A}$ has n rows and one column for each arc in $\tilde{\mathcal{A}}$ and one column for each variable f_s,

$$\mathbf{A} = [\tilde{\mathbf{A}}\ \mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_t]$$

where $\tilde{\mathbf{A}}$ is the node-arc incidence matrix of $\tilde{\mathcal{G}}$. First, we observe that the maximum rank of $\mathbf{A}$ is $n - m$ because adding all the rows up yields the zero vector for each disjoint subnetwork k corresponding to node set $\mathcal{N}_k$. Furthermore, we assume without loss of generality that each network $\tilde{\mathcal{G}}_k$ contains at least one spanning tree since otherwise we can add artificial arcs with sufficiently large costs. This implies that the rank of $\tilde{\mathbf{A}}$ is $\sum_k (n_k - 1) = n - m$ and thus $rank(\mathbf{A}) = n - m$. □

With Lemma 1, we have showed that basic feasible solutions of the linear problem P2 consist of $n - m$ basic variables whose corresponding vectors in $\mathbf{A}$ are linearly independent and the rest of the variables are fixed at their lower or upper bound. Following this fact, if none of the variables $\{f_1, \dots, f_t\}$ are in the basis, then this basis can be represented by an *m-spanning forest* in $\tilde{\mathscr{G}}$ in order to get $n - m$ linearly independent vectors. Otherwise, if r of these variables $\{f_1, \dots, f_r\}$ are basic, then we should select $n - r - m$ variables $\{f_{ij}, (i, j) \in \tilde{\mathcal{A}}\}$ whose associated vectors in the node-arc incidence matrix $\tilde{\mathbf{A}}$ are linearly independent and also independent of variables $\{f_1, \dots, f_r\}$. This latter case can be obtained by removing r arcs from an m-spanning forest in $\tilde{\mathscr{G}}$, which will decompose it into $r + m$ node-disjoint trees $\mathscr{T}_1(\mathcal{N}_1^T, \mathcal{A}_1^T), \dots, \mathscr{T}_{r+m}(\mathcal{N}_{r+m}^T, \mathcal{A}_{r+m}^T)$. This collection of trees will again span $\tilde{\mathscr{G}}$ and thus the resulting forest is a $(r + m)$-spanning forest in $\tilde{\mathscr{G}}$ which we will denote as $\mathscr{F}$. We will now analyze the structure of the related bases.

Let $\tilde{\mathbf{B}}$ denotes the submatrix of $\tilde{\mathbf{A}}$ associated with the $(r + m)$-spanning forest $\mathscr{F}$ and $\mathbf{a}_1, \dots, \mathbf{a}_r$ be vectors associated with variables $f_1, \dots, f_r$. Given that the rank of $\tilde{\mathbf{B}}$ is equal to $n-r-m$, the rank of $\mathbf{B} = [\tilde{\mathbf{B}}\ \mathbf{a}_1\ \mathbf{a}_2\ \dots\ \mathbf{a}_r]$ is equal to $n-m$ if the vectors in $\tilde{\mathbf{B}}$ with vectors $\mathbf{a}_1, \dots, \mathbf{a}_r$ are linearly independent. Accordingly, we provide in the following a suitable condition that guarantees $\mathbf{B}$ is a basis of the problem P2. But before going into the details, we need to introduce some additional notation. Lets assume that $\mathscr{T}_1, \dots, \mathscr{T}_{z_1} \subset \tilde{\mathscr{G}}_1, \mathscr{T}_{z_1+1}, \dots, \mathscr{T}_{z_2} \subset \tilde{\mathscr{G}}_2$ and in general $\mathscr{T}_{z_{(k-1)}+1}, \dots, \mathscr{T}_{z_k} \subset \tilde{\mathscr{G}}_k$ for all $k \in \mathcal{K}$. Therefore the number of spanning trees in each subnetwork $\tilde{\mathscr{G}}_k$ is equal to $z_k - z_{(k-1)}$ with $z_0 = 0$ and $z_m = r + m$. In a similar fashion, let $\mathcal{N}_1^T = \{1, \dots, n_1^T\}, \mathcal{N}_2^T = \{n_1^T + 1, \dots, n_2^T\}$, and in general $\mathcal{N}_z^T = \{n_{(z-1)}^T + 1, \dots, n_z^T\}$ for all $z \in \mathcal{Z} = \{1, \dots, z_1, z_1 + 1, \dots, z_2, \dots, z_{(m-1)} + 1, \dots, z_m\}$ without loss of generality. Finally, let $\mathbf{D}'$ be the matrix formed by the elements $d_{z,s} = \sum_{l \in \mathcal{N}_z^T} a_s^l$ for all $z \in \mathcal{Z}' = \mathcal{Z} \backslash \{z_1, z_2, \dots, z_m\}$ and $s \in \mathcal{S}' = \{1, \dots, r\}$. Note that $|\mathcal{Z}| = r + m$ and $|\mathcal{Z}'| = r$ by definition.

Theorem 1 *rank* $(\mathbf{B}) = n - m$ *if and only if rank* $(\mathbf{D}') = r$.

Proof After column arrangements, the matrix $\mathbf{B}$ can be reexpressed as

$$\mathbf{B} = \left(\begin{array}{c|c|c|c|c|c|c|c} \mathbf{T}_1 & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \Delta_1 \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{T}_{z_1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \Delta_{z_1} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{T}_{z_{(m-1)}+1} & \cdots & \mathbf{0} & \Delta_{z_{(m-1)}+1} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{T}_{z_m} & \Delta_{z_m} \end{array}\right) \tag{10}$$

where $\mathbf{0}$ are matrices of conformal dimensions with all entries equal to zero, $\mathbf{T}_z$ is the node-arc incidence matrix of $\mathscr{T}_z$ and

$$\Delta_z = \begin{pmatrix} a_1^{n_{(z-1)+1}} & a_2^{n_{(z-1)+1}} & \cdots & a_r^{n_{(z-1)+1}} \\ a_1^{n_{(z-1)+2}} & a_2^{n_{(z-1)+2}} & \cdots & a_r^{n_{(z-1)+2}} \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{n_z} & a_2^{n_z} & \cdots & a_r^{n_z} \end{pmatrix}$$

for all $z \in \mathcal{Z}$. Here, $\text{rank}(\mathbf{T}_1) = n_1^T - 1$, $\text{rank}(\mathbf{T}_2) = n_2^T - n_1^T - 1$, and in general $\text{rank}(\mathbf{T}_z) = n_z^T - n_{(z-1)}^T - 1$ for all $z \in \mathcal{Z}$. As every non-singular square submatrix of the node-arc incidence matrix of a directed network is triangular, the matrix $\mathbf{B}$ can be rewritten as

$$\mathbf{B}' = \left(\begin{array}{c|c|c|c|c|c|c|c} \mathbf{T}'_1 & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \Delta'_1 \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{T}'_{z_1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \Delta'_{z_1} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{T}'_{z_{(m-1)}+1} & \cdots & \mathbf{0} & \Delta'_{z_{(m-1)}+1} \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{T}'_{z_m} & \Delta_{z'_m} \end{array}\right) \tag{11}$$

where

$$\mathbf{T}'_z = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \pm 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \pm 1 \end{pmatrix} \quad \text{and} \quad \Delta'_z = \begin{pmatrix} d_{z,1} & d_{z,2} & \cdots & d_{z,r} \\ a_1^{n_{(z-1)+2}} & a_2^{n_{(z-1)+2}} & \cdots & a_r^{n_{(z-1)+2}} \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{n_z} & a_2^{n_z} & \cdots & a_r^{n_z} \end{pmatrix}$$

for all $z \in \mathcal{Z}$. Hence, $\text{rank}(\mathbf{B}) = \sum_{k=1}^{m} \sum_{z=z_{(k-1)}+1}^{z_k} \text{rank}(\mathbf{T}_z) + \text{rank}(\mathbf{D}) = n - r - m + \text{rank}(\mathbf{D})$ where $\mathbf{D}$ is formed by the elements $d_{z,s} = \sum_{l \in \mathcal{N}_z^T} a_s^l$ for all $z \in \mathcal{Z}$ and $s \in \mathcal{S}'$. As we have $0 = \sum_{l \in \mathcal{N}_k} a_s^l = \sum_{l \in \mathcal{N}^T_{z_{(k-1)}+1}} a_s^l + \cdots + \sum_{l \in \mathcal{N}^T_{z_k}} a_s^l$ for all $k \in \mathcal{K}$, $s \in \mathcal{S}$ from (5), $\text{rank}(\mathbf{B}) = n - m$ holds if and only if $\text{rank}(\mathbf{D}') = r$. □

Therefore, a basic solution to MCPFD consists of an $(r+m)$-spanning forest $\mathscr{F}$ in $\tilde{\mathscr{G}}$ where $r = 0, \ldots, t$ as well as variables $\{f_1, \ldots, f_r\}$ verifying that $\text{rank}(\mathbf{D}') = r$. Note that for each $(r+m)$-spanning forest, there are $\binom{t}{r}$ combinations of selecting r variables among $\{f_1, \ldots, f_t\}$.

Definition 1 An $(r+m)$-spanning forest $\mathscr{F}$ in $\tilde{\mathscr{G}}$ is a good $(r+m)$-forest with respect to the variables $\{f_s\}_{s \in \mathcal{S}'}$ with $\mathcal{S}' \subseteq \mathcal{S}$ and $|\mathcal{S}'| = r$, if $\text{rank}(\mathbf{D}') = r$ where $\mathbf{D}'$ is formed by the elements $d_{z,s} = \sum_{l \in \mathcal{N}_z^T} a_s^l$ for all $z \in \mathcal{Z}'$ and $s \in \mathcal{S}'$ and $\mathcal{N}_z^T$ is the node set of tree $\mathscr{T}_z$ in forest $\mathscr{F}$.

Theorem 2 *A basic solution of MCPFD is constituted by an $(r+m)$-spanning forest $\mathscr{F}$ in $\tilde{\mathscr{G}}$ where $r = 0, \ldots, t$ plus a set of r variables $\{f_s\}_{s \in \mathcal{S}'}$, $\mathcal{S}' \subseteq \mathcal{S}$, $|\mathcal{S}'| = r$ verifying that $\mathscr{F}$ is a good $(r+m)$-forest with respect to $\{f_s\}_{s \in \mathcal{S}'}$.*

Proof It is clear from the preceding developments. □

3 Network simplex based algorithm

In this section, we give in details the main steps required for the primal simplex algorithm developed to solve problem $P2$.

1. *Finding the initial basic feasible solution*: If none is conveniently available, then all artificial start method [10,1,15,2] can be used to get a basic feasible solution with artificial variables in the network $\tilde{\mathscr{G}}$. The initial basic feasible solution is constituted by the good m-forest defined by this feasible solution. All other variables are non-basic variables and are equal to their lower bounds or upper bounds.
2. *Computing the values of the basic variables*: From now on we assume that the basis is given by a good $(r+m)$-forest such that $\mathscr{T}_z \subset \mathscr{F}$ for all $z \in \mathcal{Z}$ and the variables $\{f_1, \ldots, f_r\}$. Let $\mathcal{B}$ be the set of arcs $(i, j) \in \tilde{\mathcal{A}}$ such that f_{ij} is a basic variable, and $\mathcal{B}'$ be the set of $s \in \mathcal{S}$ such that f_s is a basic variable. Accordingly, we categorize non-basic variables such that $\mathcal{L} = \{(i, j) \in \tilde{\mathcal{A}} \backslash \mathcal{B} : f_{ij} = 0\}$, $\mathcal{L}' = \{s \in \mathcal{S} \backslash \mathcal{B}' : f_s = 0\}$, $\mathcal{U} = \{(i, j) \in \tilde{\mathcal{A}} \backslash \mathcal{B} : f_{ij} = u_{ij}\}$ and $\mathcal{U}' = \{s \in \mathcal{S} \backslash \mathcal{B}' : f_s = u_s\}$. Finally, we let $\mathcal{V}_z^1 = \{(i, j) \in \mathcal{U} : i \in \mathcal{N}_z^T, j \notin \mathcal{N}_z^T\}$ and $\mathcal{V}_z^2 = \{(i, j) \in \mathcal{U} : i \notin \mathcal{N}_z^T, j \in \mathcal{N}_z^T\}$ for all $z \in \mathcal{Z}'$. Then, the following Theorem guarantees that the values of variables $\{f_1, \ldots, f_r\}$ are solvable.

Theorem 3 *The values of basic variables $\{f_1, \ldots, f_r\}$ are the solution of the following linear system:*

$$\mathbf{D}'\mathbf{f} = \mathbf{b}' \tag{12}$$

where $\mathbf{D}'$ is previously defined, $\mathbf{f} = (f_1, \ldots, f_r)^t$ and $\mathbf{b}' = \left(b'_1, \ldots, b'_r\right)^t$ with

$$b'_z = \sum_{l \in \mathcal{N}_z^T} b_l - \left(\sum_{(i,j) \in \mathcal{V}_z^1} u_{ij} - \sum_{(i,j) \in \mathcal{V}_z^2} u_{ij} \right) - \sum_{l \in \mathcal{N}_z^T} \sum_{s \in \mathcal{U}'} a_s^l u_s \qquad z \in \mathcal{Z}'$$

Proof After fixing the values of non-basic variables, each constraints in (7) can be reformulated as

$$\sum_{j:(i,j) \in \mathcal{B}} f_{ij} - \sum_{j:(j,i) \in \mathcal{B}} f_{ji} + \sum_{s \in \mathcal{B}'} a_s^l f_s = \hat{b}_l \qquad l \in \mathcal{N}$$

where $\hat{b}_l = b_l - \sum_{j:(i,j) \in \mathcal{U}} u_{ij} + \sum_{j:(j,i) \in \mathcal{U}} u_{ij} - \sum_{s \in \mathcal{U}'} a_s^l u_s$. Since u_{ij} vanishes if $i \in \mathcal{N}_z^T$, $j \in \mathcal{N}_z^T$ and $(i, j) \in \mathcal{U}$,

$$\begin{aligned}
\sum_{l \in \mathcal{N}_z^T} \hat{b}_l &= \sum_{l \in \mathcal{N}_z^T} b_l - \sum_{\substack{i \in \mathcal{N}_z^T,\, j \notin \mathcal{N}_z^T, \\ (i,j) \in \mathcal{U}}} u_{ij} + \sum_{\substack{i \notin \mathcal{N}_z^T,\, j \in \mathcal{N}_z^T, \\ (i,j) \in \mathcal{U}}} u_{ij} - \sum_{l \in \mathcal{N}_z^T} \sum_{s \in \mathcal{U}'} a_s^l u_s \\
&= \sum_{l \in \mathcal{N}_z^T} b_l - \sum_{(i,j) \in \mathcal{V}_z^1} u_{ij} + \sum_{(i,j) \in \mathcal{V}_z^2} u_{ij} - \sum_{l \in \mathcal{N}_z^T} \sum_{s \in \mathcal{U}'} a_s^l u_s = b'_z
\end{aligned}$$

for all $z \in \mathcal{Z}'$. Similar to the proof of Theorem 1, we may solve the linear system (12) to obtain the value of variables $\{f_1, \ldots, f_r\}$. Hence, the proof is complete. □

The values of basic variables $\{f_1, \ldots, f_r\}$ affect the requirement of each supply and demand node. Then, the flow values of the remaining arcs in each tree $\mathcal{T}_z$ $z \in \mathcal{Z}$ can be determined by applying the general procedure of the network simplex algorithm.

3. *Computing node potentials for a given basis*: Given a basic feasible solution, we have to verify if it is optimal by calculating node potentials $\boldsymbol{\pi} = (\pi_i : i \in \mathcal{N})$ and taking into account the fact that the reduced cost of each basic variable is zero. In other words, we should be able to find node potentials such that $c_{ij}^{\boldsymbol{\pi}} = 0$ for all $(i, j) \in \mathcal{B}$ and $c_s^{\boldsymbol{\pi}} = 0$ for all $s \in \mathcal{B}'$ where $c_{ij}^{\boldsymbol{\pi}} = c_{ij} - \pi_i + \pi_j$ for all $(i, j) \in \tilde{\mathcal{A}}$ and $c_s^{\boldsymbol{\pi}} = c_s - \sum_{i \in \mathcal{N}} a_s^i \pi_i$ for all $s \in \mathcal{S}$. The first condition can be satisfied by computing appropriate node potentials as in the network simplex algorithm. If these node potentials also satisfy the second condition then we are done. Otherwise, new node potentials $\tilde{\pi}$ can be calculated such that

$$\tilde{\pi}_i = \begin{cases} \pi_i + \sigma_z & \text{for all } i \in \mathcal{N}_z^T \quad \text{and} \quad z \in \mathcal{Z}' \\ \pi_i & \text{for all } i \in \mathcal{N}_z^T \quad \text{and} \quad z \in \{z_1, z_2, \ldots, z_m\} \end{cases} \tag{13}$$

where $\boldsymbol{\sigma} = \left(\sigma_z : z \in \mathcal{Z}'\right)^t$ are obtained by solving the linear system $\left(\mathbf{D}'\right)^t \boldsymbol{\sigma} = \mathbf{c}^{\boldsymbol{\pi}}$ given $\mathbf{c}^{\boldsymbol{\pi}} = (c_z^{\boldsymbol{\pi}} : z \in \mathcal{Z}')^t$ and $\mathbf{D}'$ previously defined.

Lemma 2 *The node potentials $\tilde{\boldsymbol{\pi}}$ given in (13) satisfies $c_{ij}^{\tilde{\boldsymbol{\pi}}} = 0$ for all $(i, j) \in \mathcal{B}$ and $c_s^{\tilde{\boldsymbol{\pi}}}$ for all $s \in \mathcal{B}'$.*

Proof It can be verified that node potentials $\tilde{\boldsymbol{\pi}}$ satisfy $c_{ij}^{\tilde{\boldsymbol{\pi}}} = 0$ for each $(i, j) \in \mathcal{B}$. Then, for all $s \in \mathcal{B}'$, it holds that $c_s^{\tilde{\boldsymbol{\pi}}} = c_s - \sum_{l \in \mathcal{N}} a_s^l \tilde{\pi}_i = c_s^{\boldsymbol{\pi}} - \sum_{z \in \mathcal{Z}'} \sum_{l \in \mathcal{N}_z^T} a_s^l \sigma_z = c_s^{\boldsymbol{\pi}} - \sum_{z \in \mathcal{Z}'} d_{z,s} \sigma_z = c_s^{\boldsymbol{\pi}} - c_s^{\boldsymbol{\pi}} = 0$ by taking into account the linear system given. □

4. *Testing optimality and selecting the entering variable*: Since problem P2 in (6)–(9) is a linear program, the optimality conditions can be written as

$$c_{ij}^{\boldsymbol{\pi}} \geqslant 0 \ \ (i, j) \in \mathcal{L}, \qquad c_{ij}^{\boldsymbol{\pi}} \leqslant 0 \ \ (i, j) \in \mathcal{U}, \tag{14}$$

and

$$c_s^{\boldsymbol{\pi}} \geqslant 0 \ \ s \in \mathcal{L}', \qquad c_s^{\boldsymbol{\pi}} \leqslant 0 \ \ s \in \mathcal{U}'. \tag{15}$$

If the given basis satisfies the optimality conditions (14) and (15), it is optimal and the algorithm terminates. Otherwise, the algorithm selects a non-basic variable f_{ij} with $(i, j) \in \mathcal{L} \cup \mathcal{U}$ violating the condition in (14) or a non-basic variable f_s with $s \in \mathcal{L}' \cup \mathcal{U}'$ violating the condition in (15) as entering variable according to any usual rules [1].

5. *Selecting the leaving variable*: Suppose that we have selected an entering non-basic variable which is equal to its lower bound. Increasing the value of this variable by θ units will necessitate to alter the values of some basic variables to maintain the feasibility. If the value of the entering variable hits its upper bound while the values of the modified basic variables stay between their respective bounds, then the entering variable still remains non-basic. Otherwise, the non-basic variable enters the basis and one of the basic variables will leave at its lower or upper bound. Similar arguments can be made if the entering non-basic variable is initially at its upper bound. Depending on the non-basic variable that enter the basis, we will consider three different cases to identify the leaving basic variable.

Case 1. *The entering variable is* f_{ij} *with* $i \in \mathcal{N}_z^T$ *and* $j \in \mathcal{N}_z^T$. In this case, the variable f_{ij} only affects tree $\mathcal{T}_z$. The arc corresponding to this variable is added to $\mathcal{T}_z$ which creates a unique cycle. The amount of flow θ is increased and sent through this cycle until the variable corresponding to one of the arcs of the cycle reaches its upper or lower bound. If this variable is f_{ij}, then it remains non-basic. Otherwise, it enters the basis and one of the basic variables at its lower or upper bound will leave. In either cases, the value of all basic variables corresponding to the remain arcs of the cycle are adjusted with respect to this additional amount of flow.

Case 2. *The entering variable is* f_{ij} *with* $i \in \mathcal{N}_z^T$, $j \in \mathcal{N}_{z'}^T$ *and* $z \neq z'$. Suppose that an additional amount of flow θ is sent through arc (i, j). Then, the demand of tree $\mathcal{T}_z$ decreases in θ units and the demand of $\mathcal{T}_{z'}$ increases in θ units. Therefore, new values of variables $f_1, \ldots, f_r$ are obtained by solving a modification of system (12) such that $\mathbf{D}'\mathbf{f} = \bar{\mathbf{b}}'$ where $\bar{\mathbf{b}}' = \left(b_1', \ldots, b_z' - \theta, \ldots, b_{z'}' + \theta, \ldots, b_r'\right)^t$. Once these values are determined, the value of θ is increased until f_{ij} or one of the basic variables reaches one of its bounds. Then, the arguments presented in Case 1 remains also valid here.

Case 3. *The entering variable is* f_s *where* $r < s \leqslant t$. When the value of variable f_s is increased by $\theta \geqslant 0$, the demand of each tree $\mathcal{T}_z$ for which $\sum_{l \in \mathcal{N}_z^T} a_s^l > 0$ should decrease in $\theta \sum_{l \in \mathcal{N}_z^T} a_s^l$, the demand of each tree $\mathcal{T}_z$ for which $\sum_{l \in \mathcal{N}_z^T} a_s^l < 0$ should increase in $\theta \left(-\sum_{l \in \mathcal{N}_z^T} a_s^l\right)$. Therefore, the new values of variables $f_1, \ldots, f_r$ are determined by solving a modification of system (12) such that $\mathbf{D}'\mathbf{f} = \bar{\mathbf{b}}'$ where $\bar{\mathbf{b}}' = \left(b_1' - \theta \sum_{l \in \mathcal{N}_1^T} a_s^l, \ldots, b_r' - \theta \sum_{l \in \mathcal{N}_r^T} a_s^l\right)^t$. Then, the arguments presented in Case 1 remains also valid here.

4 Conclusion

In this study, we investigated a variant of MCNF problem which we briefly called the minimum cost proportional flow problem with disconnected networks. The proposed model covers the equal flow problem, some generalized network flow problems and the manufacturing network flow problem. After reformulating the problem, we have shown that the bases are characterized as good $(r+m)$-forests which are very similar to spanning trees. Based on this property, we developed a primal simplex algorithm that exploits the network structure of the problem and requires only slight modifications of the network simplex algorithm.

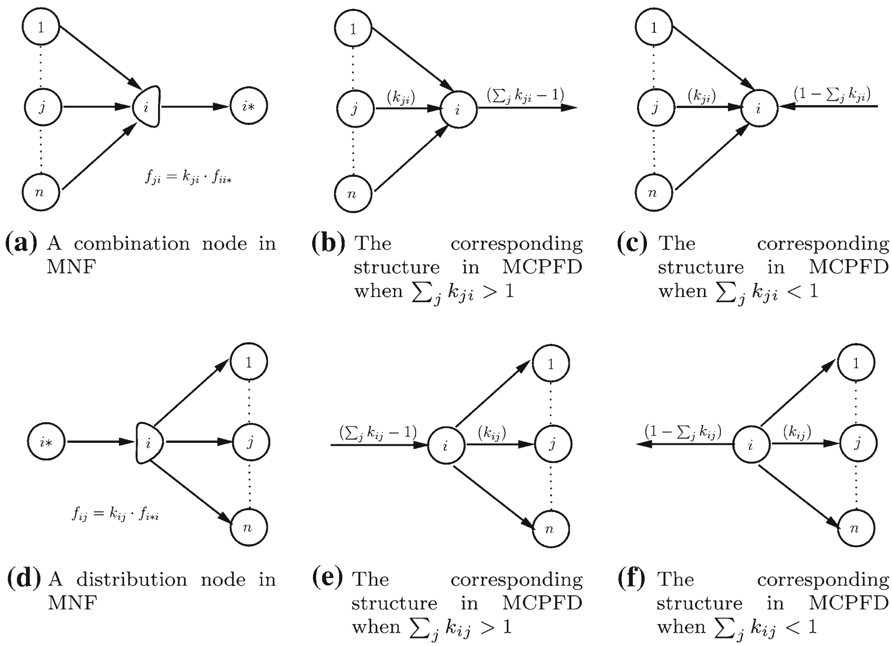
We showed that the instances of MCPFD and MNF problems can be transformed to each other. However, the algorithm given in this study to solve the MCPFD problem has some important aspects that distinguish it from other existing ones developed to solve the MNF problem. As for example, the detachment procedure proposed by Wang and Lin [17] partitions the basis into $t + 1$ basic components, each corresponding to a tree, at the very start. However, our solution algorithm includes at most this amount of complicating variables into the basis and only when they are needed. It thus has the possibility to better exploit the embedded network structure. Moreover, the algorithm is tailored to deal directly with the decomposable structure of the MCPFD problem. In another case, Lu et al. [12] presented an extended cycle method to calculate the reduced costs for non-basic variables when solving the MNF problem. Note that our algorithm calculates the reduced costs for non-basic variables based on the node potentials at the cost of solving r linear equations, which is more time efficient.

The application of the method will be especially consequential for large scale problems (e.g. inventory planning of a product assembled from several modules each of which supplied from separate logistics networks) and for problems where MCPFD is included as a subproblem and must be solved within an algorithmic framework several times.

Acknowledgments This work has been financially supported by Galatasaray University Research Fund under Grant No.10.402.007.

5 Appendix

In Fig. 2, the transformations of two special nodes in MNF, namely the combination and distribution nodes, to the equivalent structures in MCPFD are illustrated. Here



(a) A combination node in MNF

(b) The corresponding structure in MCPFD when $\sum_j k_{ji} > 1$

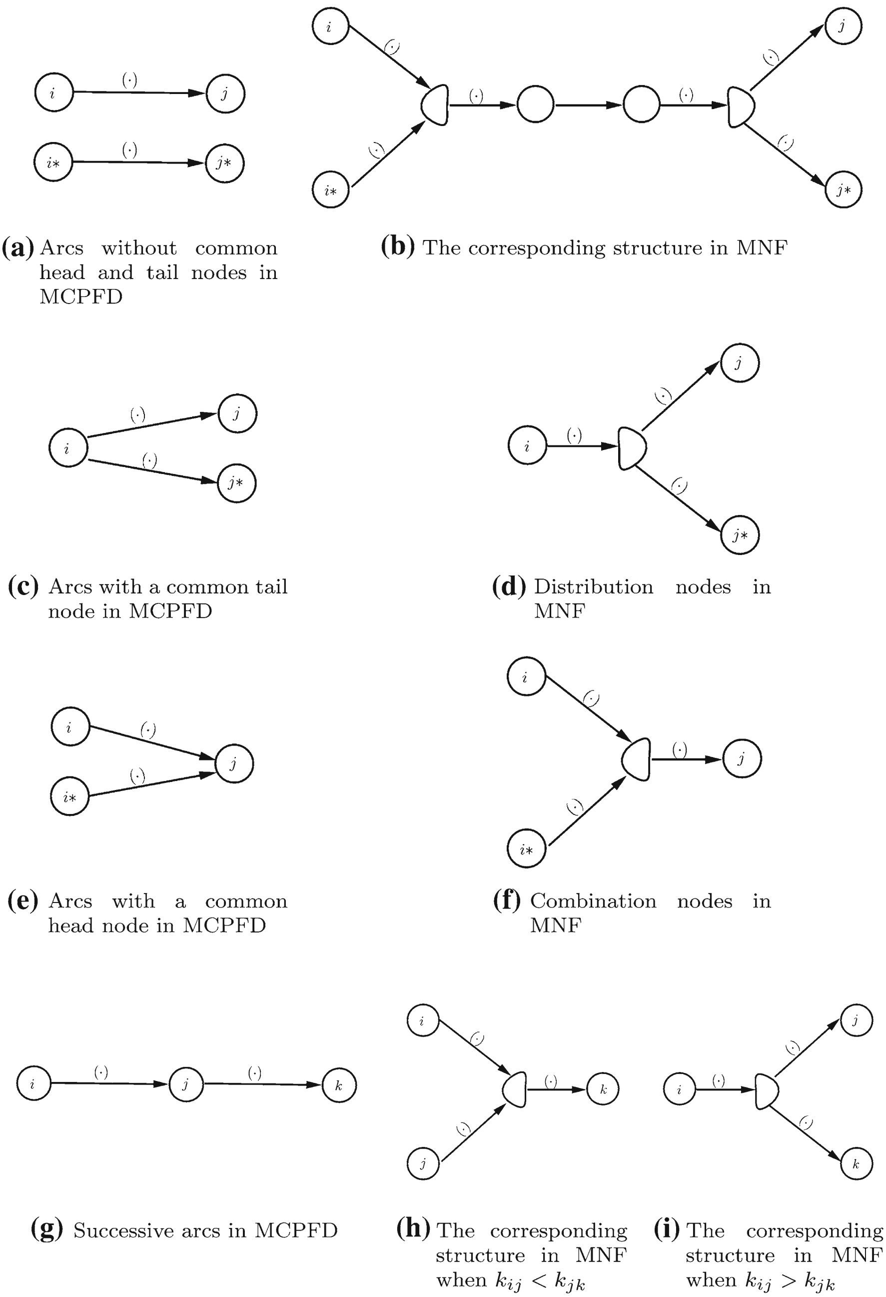
(c) The corresponding structure in MCPFD when $\sum_j k_{ji} < 1$

(d) A distribution node in MNF

(e) The corresponding structure in MCPFD when $\sum_j k_{ij} > 1$

(f) The corresponding structure in MCPFD when $\sum_j k_{ij} < 1$

Fig. 2 Transformation of combination and distribution nodes in MNF to the equivalent structures in MCPFD



(a) Arcs without common head and tail nodes in MCPFD

(b) The corresponding structure in MNF

(c) Arcs with a common tail node in MCPFD

(d) Distribution nodes in MNF

(e) Arcs with a common head node in MCPFD

(f) Combination nodes in MNF

(g) Successive arcs in MCPFD

(h) The corresponding structure in MNF when $k_{ij} < k_{jk}$

(i) The corresponding structure in MNF when $k_{ij} > k_{jk}$

Fig. 3 Transformation of different cases in MCPFD to the equivalent structures in MNF

k_{ij} and k_{ji} correspond to the proportionality coefficients. There is no need to devise any other transformations for other special nodes in MNF to obtain equivalent structures in MCPFD. In Fig. 3, the transformations of different network structures with proportional flow requirements in MCPFD to the equivalent structures in MNF are provided. In each subfigure, the sign $(\cdot)$ over any arc implies that the flow on that arc must be proportional to some other arc flow in the same figure.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
2. Ahuja, R.K., Orlin, J.B., Sechi, G.M., Zuddas, P.: Algorithms for the simple equal flow problem. Manag. Sci. **45**(10), 1440–1455 (1999)
3. Ali, A.I., Kennington, J., Shetty, B.: The equal flow problem. Eur. J. Oper. Res. **36**(1), 107–115 (1988)
4. Bahçeci, U., Feyzioğlu, O.: Inventory allocation in multi-period multi-echelon logistics network of modular products. In: Proceedings of the 8th ENIM IFAC International Conference of Modeling and Simulation, vol. 3, pp. 1021–1030. Hammamet, Tunisia (2010)
5. Calvete, H.I.: Network simplex algorithm for the general equal flow problem. Eur. J. Oper. Res. **150**(3), 585–600 (2003)
6. Du, D.Z., Pardalos, P.M.: Network Optimization Problems: Algorithms, Applications and complexity. Applied Mathematics. World Scientific, Singapore (1993)
7. Eksioglu, S.D.: Inventory Management in Supply Chains. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, vol. 2., pp. 1766–1770. Springer, Berlin (2009)
8. Fang, S.C., Qi, L.Q.: Manufacturing network flows: a generalized network flow model for manufacturing process modelling. Optim. Methods Softw. **18**(2), 143–165 (2003)
9. Helgason, R.V., Kennington, J.L.: Primal simplex algorithms for minimum cost network flows. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Handbooks in Operations Research and Management Sciences: Network Models, vol. 8., pp. 85–133. Elsevier, Amsterdam (1995)
10. Kennington, J.L., Helgason, R.V.: Algorithms for Network Programming. Wiley, New York (1980)
11. Lu, H., Yao, E., Qi, L.: Some further results on minimum distribution cost flow problems. J. Combin. Optim. **11**(4), 351–371 (2006)
12. Lu, H.Y., Yao, E.Y., Zhang, B.W.: A note on a generalized network flow model for manufacturing process. Acta Math. Appl. Sin. Engl. Ser. **25**(1), 51–60 (2009)
13. Mo, J., Qi, L., Wei, Z.: A manufacturing supply chain optimization model for distilling process. Appl. Math. Comput. **171**(1), 464–485 (2005)
14. Mo, J., Qi, L., Wei, Z.: A network simplex algorithm for simple manufacturing network model. J. Ind. Manag. Optim. **1**(2), 251–273 (2005)
15. Pardalos, P.M., Hearn, D.W., Hager, W.W.: Network Optimization, Lecture Notes in Economics and Mathematical Systems, vol. 450. Springer, Berlin (1997)
16. Venkateshan, P., Mathur, K., Ballou, R.H.: An efficient generalized network-simplex-based algorithm for manufacturing network flows. J. Combin. Optim. **15**(4), 315–341 (2008)
17. Wang, I.L., Lin, S.J.: A network simplex algorithm for solving the minimum distribution cost problem. J. Ind. Manag. Optim. **5**(4), 929–950 (2009)