Operations Research Letters 49 (2021) 559–564

Contents lists available at ScienceDirect

Operations Research Letters

www.elsevier.com/locate/orl

Complexity, algorithms and applications of the integer network flow with fractional supplies problem

Dorit S. Hochbaum [a], Asaf Levin [b,*], Xu Rao [a]

[a] Department of IEOR, Etcheverry Hall, Berkeley, CA, United States of America
[b] Faculty of Industrial Engineering and Management, The Technion, Haifa, Israel

ARTICLE INFO

Article history:
Received 4 April 2021
Received in revised form 31 May 2021
Accepted 2 June 2021
Available online 8 June 2021

Keywords:
Network flow
Proportional imbalance
Bounded total imbalance
Fractional supplies

ABSTRACT

We consider here the integer minimum cost network flow when some of the supplies are fractional. In the presence of fractional supplies it is impossible to satisfy the flow balance constraints, creating an *imbalance*. We present here a polynomial time algorithm for minimizing the total cost of flow and imbalance penalty. We also show that in the presence of a constraint that bounds the imbalance the problem is NP-hard, but efficiently solvable for a fixed number of fractional supplies.

1. Introduction

We consider here a problem of integer minimum cost network flow when some of the supplies and demands are fractional. In the presence of fractional supplies it is impossible to satisfy the flow balance constraints at the nodes of the graph. The extent to which the supplies or demands are violated is called the imbalance. The goal is to minimize the cost of the flow while minimizing the imbalance as well. The application motivating our study of this problem is the minimization of proportional imbalance of covariates (see e.g. [12,7]). In [7] it is shown that this covariates problem, for two covariates, is solved in polynomial time as a feasible integer flow on a certain network that minimizes the total imbalance at the supplies and demands nodes. The proportional imbalance problem has the supplies and demands assuming, in general, non-integral values. Therefore at such non-integral supply nodes the flow balance constraints must be violated and create imbalance.

Here we introduce a general problem of a minimum cost integer flow problem where there is a penalty to the imbalance. The goal is to minimize the sum of the total flow costs and the imbalance penalty. This problem is particularly challenging when the supplies and demands may assume non-integral values, yet the flows are required to be integers. A further generalization of the problem includes a different imbalance penalty for each node. We call this generalized problem the *fractional supplies min cost network flow* problem (FS-MCNF).

The FS-MCNF problem we address can be viewed within the framework of "optimally" restoring feasibility. Given a constrained optimization problem, questions on modifying constraints or identifying minimal subset of constraints the removal of which will restore feasibility, have been studied extensively, see e.g. [2–4,10]. The constraints' modification models typically address linear programming problems for which changes in the right hand sides will restore feasibility. One objective function studied is to minimize the sum of the changes to the right hand sides, [2,10]. In [2] changes to the constraints' coefficients and additional objective functions, such as minimizing the maximum change or deviation are also considered. The identification of subsets of constraints to remove so that the resulting problem would be feasible, are addressed in [3,4]. This latter type of problem is considered, not only for linear programming, but also for integer programming and nonlinear programming. This type of problem however is generally NP-hard, and is usually solved with heuristic techniques. The problem we address here is an integer programming problem for which the set of infeasibility-causing constraints is trivial to identify: these are the constraints for supply/demand nodes with fractional (non-integer) supplies. The method of "restoring feasibility" is indeed changes to the right hand sides, with an objective function that is the original objective plus a weighted sum of the modifications (which in our context are "imbalances"). To the best of our knowledge, such questions for integer programming with fractional right hand sides are addressed here for the first time.

[*] Corresponding author.
E-mail addresses: hochbaum@ieor.berkeley.edu (D.S. Hochbaum), levinas@technion.ac.il (A. Levin), xrao@berkeley.edu (X. Rao).

https://doi.org/10.1016/j.orl.2021.06.003

Our main contribution here is to show that FS-MCNF is solvable in polynomial time, whereas the problem variant which seeks to minimize the cost of the flow subject to *bounded imbalance* (or bounded imbalance penalty), BI-MCNF, is shown here to be NP-hard, even if all the penalties are equal to 1. Furthermore, for the case when the number of fractional supply nodes is constant we devise here a polynomial time algorithm.

To formalize the description of the problems addressed here we recall the definition of the *minimum cost network flow problem* (MCNF) (see e.g. [1,9]). The input to MCNF is a graph $G = (V, A)$ with a set of nodes V and a set of arcs A, where each arc $(i, j) \in A$ is associated with a cost c_{ij}, capacity upper bound u_{ij}, and capacity lower bound l_{ij}. Each node $i \in V$ has supply $b(i)$ which is interpreted as demand if negative, and can be 0. Let x_{ij} be the amount of flow on arc $(i, j) \in A$. The flow vector $\mathbf{x}$ is said to be feasible if it satisfies:

(1) **Flow balance constraints:** For every node $k \in V$, $Outflow(k) - Inflow(k) = b(k)$.

(2) **Capacity constraints:** For each arc $(i, j) \in A$, $l_{ij} \le x_{ij} \le u_{ij}$.

The linear programming formulation of the problem is:

$$\begin{array}{lll}
\text{(MCNF)} & \min & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{subject to} & & \sum_{(k,j)\in A} x_{kj} - \sum_{(i,k)\in A} x_{ik} = b(k), \ \forall k \in V \\
 & & l_{ij} \le x_{ij} \le u_{ij}, \ \ \forall (i,j) \in A.
\end{array}$$

It is well known that the constraint matrix of MCNF is totally unimodular. This means that for integer supplies and integer capacity bounds, all the basic solutions are integral. The problem investigated here has fractional (non-integral) supplies $b(i)$, but still requires integer flows. For a node k with non-integer supply $b(k)$ it is therefore impossible to meet the respective flow balance constraint, since both outflow and inflow are integral and therefore the difference $Outflow(k) - Inflow(k)$ must be integral as well. For a flow that satisfies the capacity constraints but does not satisfy the flow balance constraints we refer to the quantity $Outflow(k) - Inflow(k) - b(k)$ as the *discrepancy* at node k. The discrepancy can be positive or negative and is referred to as *excess* and *deficit* respectively. The absolute value of the discrepancy is called the *imbalance* at node k. The total imbalance is the sum of the imbalances at all nodes. Note that imbalances may be forced due to the lack of flow that satisfies the capacity constraints, even if supplies are integers.

One problem introduced here, the minimum imbalance MCNF, is to find an integer flow that satisfies the capacity constraints and minimizes the objective function of MCNF plus the sum of penalties assigned to the imbalances at the nodes. Note that we may allow for imbalances also at 0-supply nodes and can also allow positive imbalances restricted to only a subset of the nodes. When supplies (demands) are fractional, non-integral, there will necessarily be positive imbalances at those nodes. The FS-MCNF problem is to minimize the cost of the integer flows plus the weighted imbalance penalty.

A problem that arises in the study of covariates is the covariates minimum imbalance problem, known also as Balance Optimization Subset Selection, (BOSS). The problem of BOSS, for two covariates, was shown to be solved as a certain network flow problem [11,6,7]. We note that the BOSS problem for 3 or more covariates is NP-hard (proved in [11,6,7]) and has a straightforward algorithm for a single covariate. In this two covariates imbalance problem the supplies are integral and the imbalances are forced due to capacity bounds that do not allow a feasible flow solution. In the minimum *proportional* covariates imbalance problem, in addition to the capacities that may not allow a feasible solution, also the supplies are non integral. This proportional covariates imbalance problem, was shown to be polynomial time solvable, [7], via a construction of a specific network with zero costs for the flow and the only objective is to minimize the unweighted sum of the imbalances at individual nodes. This is a special unweighted case of FS-MCNF that applies only to a specific (bipartite) network where there are no costs for the flow.

Note that in case some of the supplies/demands are fractional, then the integer program of finding a minimum cost integer flow that satisfies all constraints is infeasible. In this case our two problems could be considered as finding a modified right hand side (referred to as the perturbed right hand side) so that the resulting integer program would be feasible. Here we find such perturbed right hand side so that the sum of the cost of the flow and the weighted penalty of the perturbations will be minimized (a special case of FS-MCNF) or find such a perturbed right hand side with bounded ℓ_1 norm of the perturbation so that the cost of the optimal integer flow will be minimized (a special case of BI-MCNF).

Paper overview. In subsection 1.1 we present the covariates proportional imbalance problem that motivated this study. Section 2 provides an integer programming formulation for FS-MCNF, and a network flow formulation where a minimum cost (integer) flow provides an optimal solution to the problem. Section 3 provides the proof that the minimum cost network flow, in the presence of non-integer supplies, subject to bounded total imbalance, is NP-hard. Subsection 3.2 describes a polynomial time algorithm solving (BI-MCNF) for a constant number of penalized fractional supplies, first for the case where all supplies are integers and then for the case where the number of fractional supplies is a constant.

Notation. For every node i, we let $\lfloor b(i) \rfloor$ be the floor of $b(i)$ and $\{b(i)\} = b(i) - \lfloor b(i) \rfloor$ be the fractional part of $b(i)$.

1.1. The covariates proportional imbalance problem

Since the covariates proportional imbalance problem is the major motivation for studying FS-MCNF, we provide next some background on this problem. In observational studies, unlike randomized experimental studies, the researcher does not have the ability to determine treatment assignment and instead is only able to observe some units that were treated and some that were not. The observed treated and untreated units are called the *treatment sample* and *control sample*, respectively. In most applications, the treatment sample is smaller than the control sample. Estimating the treatment effect in an observational study is generally more difficult than in an experimental study, where the treatment sample and control sample are both drawn, randomly, from some population. A fundamental problem related to the choice of control samples that are comparable in terms of covariates to the treatment sample is called the *minimum imbalance* problem (min-imbalance).

The covariates are the features of the units that take on several labels for each covariate. The class of units with the same label for a given covariate is called the *level*. To formalize the problem let P be the number of covariates to be balanced. For the min-imbalance the goal is to find a selection of n control samples that matches as closely as possible the sizes of the levels in the treatment sample. To introduce the min-imbalance problem, consider first the case of a single covariate, $P = 1$, that partitions the treatment and control samples into, say, k levels each. Let the sizes of the treatment and control samples be n and n', respectively, and let the sizes of the levels in the treatment sample be $\ell_1, \dots, \ell_k$ and the sizes of the levels in the control sample be $\ell'_1, \dots, \ell'_k$. The min-imbalance problem is to select a subset of the control sample, called the *selection* and denoted by S, such that the proportions of units at each level in the treatment sample and the selection are as close as possible. When the size of the selection, $|S|$, is required to be the same as the size of the treatment sample n (assuming that $n \le n'$), the objective in the min-imbalance problem is given in terms of the numbers of units at each level instead of the proportion.

The solution to the single covariate min-imbalance problem with a selection of size n is trivial: for any level i, if $\ell_i \leq \ell'_i$, then the optimal selection takes any ℓ_i control units from level i; otherwise, the optimal selection takes all ℓ'_i control units from level i. After this, take enough remaining control units from any level to reach a selection size of n. Let ℓ''_i denote the number of control units in the selection at level i. Then the value of the objective function corresponding to this selection is $\sum_{i=1}^{k} \left|\ell_i - \ell''_i\right|$, which is the optimal value for the single covariate min-imbalance problem.

For the case of multiple covariates, covariate p partitions both treatment and control samples into k_p levels each. Denote the sizes of the levels in the treatment sample under covariate p by $\ell_{p,1}, \ell_{p,2}, \ldots, \ell_{p,k_p}$, and let the partition of the control sample under covariate p be $L'_{p,1}, L'_{p,2}, \ldots, L'_{p,k_p}$ of sizes $\ell'_{p,1}, \ell'_{p,2}, \ldots, \ell'_{p,k_p}$. The *min-imbalance problem* is to find a selection S of size n, which is a subset of the control sample, that minimizes the imbalance:

$$\min \sum_{p=1}^{P} \sum_{i=1}^{k_p} \left| \left| S \cap L'_{p,i} \right| - \ell_{p,i} \right| \quad \text{s.t. } |S| = n.$$

When the selection is required to have size q for $q \leq n'$ and $q \neq n$, the proportional imbalance problem is defined. Here the min-imbalance objective looks at the difference in *proportion* of units selected at each level of each covariate, and the *min-proportional imbalance problem* is,

$$\min \sum_{p=1}^{P} \sum_{i=1}^{k_p} \left| \frac{|S \cap L'_{p,i}|}{q} - \frac{\ell_{p,i}}{n} \right| \quad \text{s.t. } |S| = q.$$

In [7] it was shown that for two covariates the min-imbalance and the min-proportional imbalance problems are solved in polynomial time. Both problems for three or more covariates are shown to be NP-hard. The min-proportional imbalance problem is formulated as a specific minimum cost network flow problem with fractional supplies. This network flow problem is a special case of seeking integer flows so as to minimize the total imbalance for a vector of supplies defined based on the requested proportions.

2. Solving fractional supplies MCNF in polynomial time

In this section, we show how to transform FS-MCNF into an integer MCNF problem, and how to derive an optimal solution for FS-MCNF from the corresponding integer MCNF optimal solution in polynomial time.

We first present the mixed-integer programming formulation for FS-MCNF.

Let $G = (V, A)$ be a directed network with a cost c_{ij} and capacity bounds u_{ij} and l_{ij} associated with every arc $(i, j) \in A$. Each node $i \in V$ is associated with a number $b(i)$ which indicates its supply or demand, and a non-negative number w_i which indicates the penalty for unit of imbalance of node i. We define the decision variables as follows:

x_{ij}: the amount of flow on arc (i, j), for $(i, j) \in A$;
d_i: the deficit on node i, for $i \in V$;
e_i: the excess on node i, for $i \in V$.

Using these decision variables, we present a mixed-integer programming formulation for FS-MCNF:

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in V} w_i \cdot (d_i + e_i) \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} + d_i - e_i = b(i) \qquad \forall i \in V \tag{2.1b}$$

$$d_i, e_i \geq 0 \qquad \forall i \in V \tag{2.1c}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in A \tag{2.1d}$$

$$x_{ij} \text{ integer} \qquad \forall (i, j) \in A. \tag{2.1e}$$

Next, we present an equivalent integer programming formulation of the mixed-integer program (2.1). In order to present our equivalent integer programming formulation, we introduce another set of decision variables. For each $i \in V$ we use an additional decision variable z_i, which is an indicator variable. Furthermore, instead of d_i we will have the decision variable d'_i that is the deficit of node i beyond $\lfloor b(i) \rfloor + 1$, and instead of e_i we will use the decision variable e'_i denoting the excess of node i beyond $\lfloor b(i) \rfloor$.

For every node i we introduce a cost coefficient w'_i that will be the coefficient of z_i in the objective function of the integer program below and we set $w'_i = w_i \cdot (2\{b(i)\} - 1)$ and $z_i = 1$ would mean that node i has a positive deficit. Observe that for all i, we have $|w'_i| \leq w_i$.

Using these decision variables and constants, we consider the following integer programming formulation.

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in V} w_i \cdot \left(d'_i + e'_i\right) + \sum_{i\in V} w'_i \cdot z_i \tag{2.2a}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} + d'_i + z_i - e'_i = \lfloor b(i) \rfloor + 1 \quad \forall i \in V \tag{2.2b}$$

$$d'_i, e'_i \geq 0 \qquad \forall i \in V \tag{2.2c}$$

$$0 \leq z_i \leq 1 \qquad \forall i \in V \tag{2.2d}$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in A \tag{2.2e}$$

$$x_{ij}, d'_i, e'_i, z_i \text{ integer} \qquad \forall (i, j) \in A \text{ and } \forall i \in V. \tag{2.2f}$$

In order to prove that (2.2) is equivalent to (2.1), we denote the constant term

$$C = \sum_{i\in V} w_i \cdot (1 - \{b(i)\}).$$

We will show that if there is an optimal solution for (2.2) of cost F, then we can construct a solution to (2.1) of cost $C + F$ in polynomial time, and furthermore if we are given an optimal solution for (2.1) of cost F', then we can construct a feasible solution for (2.2) of cost $F' - C$. This will show that it suffices to find an optimal solution for (2.2).

Lemma 2.1. *Given an optimal solution for* (2.2) *of cost F, we can construct in polynomial time a solution to* (2.1) *of cost $C + F$ in polynomial time. Our construction is invertible, so if there is a solution of cost $C + F$ to* (2.1)*, then there is a solution of cost F to* (2.2)*.*

Proof. Let (x, d', e', z) be the given optimal solution for (2.2), we let the same x define the flow variables for our solution of (2.1). The other variables are defined so that constraints (2.1b) are satisfied and the assertion that for all i if $d_i > 0$ then $e_i = 0$. Observe

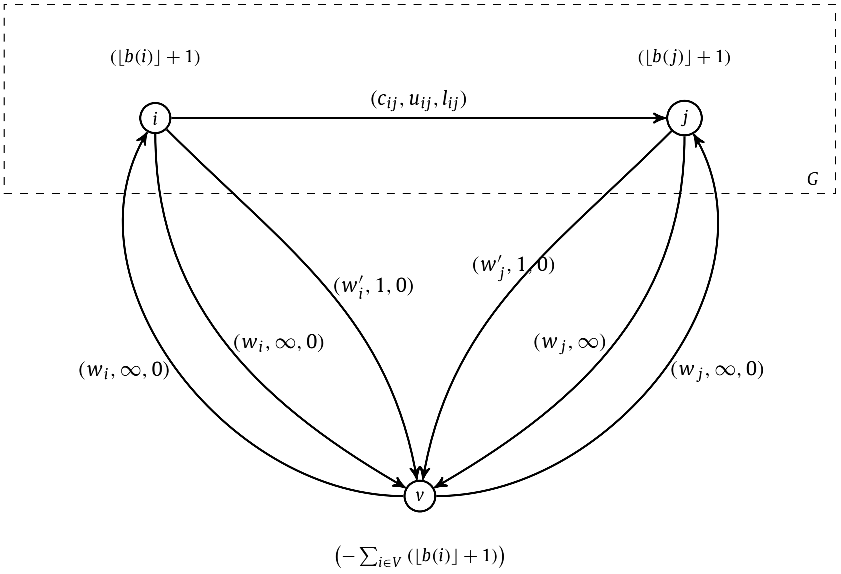
$(\lfloor b(i)\rfloor+1)$
$(\lfloor b(j)\rfloor+1)$
(c_{ij}, u_{ij}, l_{ij})
i
j
G
$(w'_i, 1, 0)$
$(w'_j, 1, 0)$
$(w_i, \infty, 0)$
(w_j, ∞)
$(w_i, \infty, 0)$
$(w_j, \infty, 0)$
v
$(-\sum_{i\in V}(\lfloor b(i)\rfloor+1))$

Fig. 1. The schematic transformed standard MCNF network in which supplies are integral and an optimal flow solution is optimal for FS-MCNF.

that there is a unique vector of (d, e) values that satisfies these conditions. When we move from (2.2) to (2.1) the right hand side of the constraints (2.2b) decreases by $1-\{b(i)\}$. We will provide assignment of values to the decision variables e_i, d_i defined below so that the left hand side of these constraints will be decreased by the same amount.

If $e'_i = 0$ and $z_i = 1$, then we let $e_i = 0$ and $d_i = d'_i + \{b(i)\}$. This assignment of variables decreases the left hand side of the constraint corresponding to i in (2.2b) from its value in (2.2) by $1-\{b(i)\}$ as required. In the other direction if $e_i = 0$, then we let $z_i = 1$, $e'_i = 0$, and $d'_i = d_i - \{b(i)\}$. Since the flow variables are integral and using $e_i = 0$ we conclude that $d_i \geq \{b(i)\}$ so $d'_i \geq 0$, and using the same arguments as for the other direction we get that the constraint corresponding to i in (2.1b) in (2.1) is satisfied if it used to hold in (2.2).

Furthermore, if $e'_i = z_i = 0$ then without loss of generality $d'_i = 0$ as otherwise we could decrease d'_i while increasing z_i without increasing the objective function value using $w'_i \leq w_i$. We let $e_i = 1-\{b(i)\}$ and $d_i = 0$ so in particular $e_i = e'_i + 1 - \{b(i)\}$. This assignment of variables decreases the left hand side of the constraint corresponding to i in (2.2b) from its value in (2.2) by $1-\{b(i)\}$ as required.

In the last remaining case assume that $e'_i \geq 1$ so we conclude that $z_i = 0$ since $w'_i \geq -w_i$. We let $e_i = e'_i + 1 - \{b(i)\}$ and $d_i = 0$. This assignment of variables decreases the left hand side of the constraint corresponding to i in (2.2b) from its value in (2.2) by $1-\{b(i)\}$ as required. In the other direction if $e_i > 0$ (so without loss of generality $d_i = 0$) and we can assume that $e_i \geq 1-\{b(i)\}$ using the integrality of the flow variables. Then we let $z_i = 0$, $e'_i = e_i - 1 + \{b(i)\}$, and $d'_i = 0$ and using the same arguments we get that the constraint corresponding to i in (2.1b) in (2.1) is satisfied if it used to hold in (2.2).

It remains to consider the costs of the two solutions. The flow x has the same cost $\sum_{(i,j)\in A} c_{ij}x_{ij}$ in both problems. We will compare the imbalance cost. It suffices to show that the *imbalance cost associated with node* i in (2.1) that is $w_i \cdot (d_i + e_i)$ is obtained from the imbalance cost associated with i in (2.2) that is $w_i \cdot \left(d'_i + e'_i\right) + w'_i \cdot z_i$ by adding the term $w_i \cdot (1-\{b(i)\})$

First assume that $z_i = 1$. Then, without loss of generality using $|w'_i| \leq w_i$, we assume that $e'_i = 0$ as otherwise the cost of the solution to (2.2) could be decreased by decreasing z_i and e'_i by 1. Then the imbalance cost associated with node i in (x, d', e', z) is $d'_i \cdot w_i + w'_i$. By adding $w_i \cdot (1-\{b(i)\})$ and using the definition of $w'_i = w_i \cdot (2\{b(i)\} - 1)$ we get that the total imbalance cost of i is $(d'_i + \{b(i)\}) \cdot w_i = d_i \cdot w_i$ as required.

Next, consider the case where $z_i = 0$. Then $d'_i = 0$ as otherwise we could decrease d'_i by 1 and increase z_i to 1 without increasing the cost of the solution. Then the imbalance cost associated with node i in (x, d', e', z) is $e'_i \cdot w_i$. By adding $w_i \cdot (1-\{b(i)\})$, we get that the total imbalance cost of i is $(e'_i + 1 - \{b(i)\}) \cdot w_i = e_i \cdot w_i$ as required. □

We conclude that in order to solve our mixed-integer program (2.1) it suffices to solve the integer program (2.2). It could be observed that the last integer program is a standard formulation of the min cost network flow problem obtained from the input network G by adding one additional node v and adding three arcs incident to each node $i \in V$. The first such arc is (v, i) of infinite capacity and cost per unit of flow w_i, the flow along this arc corresponds to the value of e'_i. The second such arc is (i, v) of infinite capacity and cost per unit of flow of w_i, the flow along this arc corresponds to the value of d'_i. The last such arc is a unit capacity arc of cost w'_i that is the arc (i, v). Each of these $3|V|$ additional arcs has a lower bound of zero on the flow along the arc. In the resulting instance, the new node v has supply of $-\sum_{i\in V}(\lfloor b(i)\rfloor + 1)$ and node $i \in V$ has supply of $\lfloor b(i)\rfloor + 1$. Fig. 1 is a schematic illustration of how to convert an FS-MCNF network G into a standard MCNF network which has $O(|V|)$ nodes and $O(|A|)$ arcs. An efficient strongly polynomial algorithm for a MCNF on n nodes and m arcs has complexity of $O(m\log n(m + n\log n))$, see Orlin [9]. Thus, we have established the following theorem.

Theorem 2.2. *Problem* FS-MCNF *is solved in polynomial time.*

3. Bounded imbalance MCNF

Here, instead of penalizing for imbalance of some nodes as we did in the last section, we consider the optimization problem resulting from upper bounding the total imbalance of all nodes. Namely we will consider the following mixed-integer program.

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} + d_i - e_i = b(i) \qquad \forall i \in V \tag{3.1b}$$

$$\sum_{i\in V}(d_i + e_i) \le B \tag{3.1c}$$

$$d_i, e_i \ge 0 \qquad \forall i \in V \tag{3.1d}$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in A \tag{3.1e}$$

$$x_{ij} \text{ integer} \qquad \forall (i,j) \in A. \tag{3.1f}$$

The constraint (3.1c) serves as a complicating constraint and it is introduced in this formulation where B is the upper bound on the total imbalance of the nodes. Observe that if $B = 0$, the problem is the standard MCNF problem as for all i $d_i = e_i = 0$ so no node has a positive imbalance. We first show that this problem, denoted as BI-MCNF is NP-hard. Then we consider special cases of this problem that turn out to be polynomial time solvable: in the first case all supplies are integer, and in the second one the number of non-integral supplies is a constant.

3.1. BI-MCNF *is NP-hard*

We consider the following NP-hard variant of the subset-sum problem [5,8], where the input consists of non-negative integers $a_1, a_2, \ldots, a_n$, a positive target value T (integer) and number k. The goal is to check if there exists a subset S of the index set $\{1, 2, \ldots, n\}$ consisting of exactly k indexes such that $\sum_{i\in S} a_i = T$. The fact that this is an NP-hard problem is folklore, and a straightforward reduction from the subset-sum problem is the following one. Given an instance for the subset-sum problem consisting of n non-negative integers and a target value T, we add n times 0 to the list of integers so it now consists of $2n$ integers and use the same target value T, let $k = n$ and conclude that the resulting instance of our variant is a YES instance if and only if the original subset-sum instance is a YES instance.

Theorem 3.1. BI-MCNF *is NP-hard.*

Proof. We present a polynomial reduction from the last variant of the subset-sum problem that we denote as P. Consider an input to P consisting of non-negative integers $a_1, a_2, \ldots, a_n$, a positive target value T (integer) and number k. First, if there is an index i for which $a_i > T$, then we can delete it from the instance of P without changing the problem as such index cannot belong to a feasible solution of the problem. Thus, without loss of generality we assume that the instance satisfies $0 \le a_i \le T$ for all i. We construct a star graph with n leaves (and one root node r) where all arcs are oriented from the root node to the leaves of the star. The capacity of every arc is 1, and the supply of the root node is k. The cost of the arc from the root to the i-th leaf is $-a_i$. In the decision variant of problem BI-MCNF the question would be if there is a solution to the problem of cost at most $-T$. It suffices to show that the decision variant of BI-MCNF is NP-complete.

We still need to define the demand of the leaves of the star and the budget on the total imbalance. We have the association that the i-th leaf is associated with the i-th integer a_i in the input to P. Our goal is to define a demand $b(i)$ for the i-th leaf satisfying that if we increase the value of the flow entering the i-th leaf from 0 to 1 then the imbalance of the i-th leaf is increased by $\frac{a_i}{4T}$. So let $b(i) = \frac{1-\frac{a_i}{4T}}{2}$, and define the budget of the total imbalance as $\frac{1}{4} + \sum_i b(i)$. This completes the description of the polynomial reduction and note that we indeed compute the instance of BI-MCNF in linear time (Fig. 2).

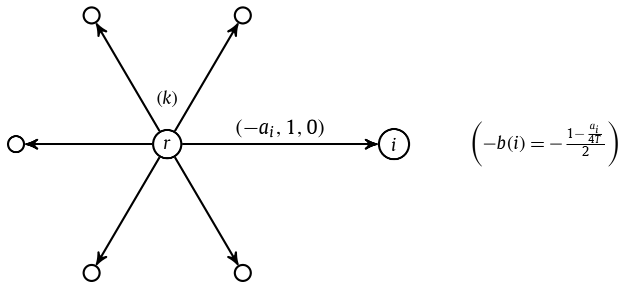

Fig. 2. The network of an equivalent BI-MCNF instance of the subset-sum problem.

First, every leaf i has an imbalance of at least $b(i)$ in every feasible (integer) solution as $0 \le b(i) \le \frac{1}{2}$. Thus, if there is a feasible solution to our instance of BI-MCNF then the root node has a zero imbalance so exactly k units of flow leave the root node. Consider a feasible solution for the decision variant of BI-MCNF of cost at most $-T$, then it corresponds to k units of flow leaving the root node, that use a collection of exactly k arcs leaving the root and reaching k leaves. These k leaves are associated with k integers in the input of P. The set of the k leaves is denoted as S. By the bound on the total imbalance we have that $\sum_{i\in S} \frac{a_i}{4T} \le \frac{1}{4}$ so $\sum_{i\in S} a_i \le T$. By the upper bound on the cost of the solution we have that $\sum_{i\in S}(-a_i) \le -T$ so $\sum_{i\in S} a_i \ge T$. We conclude that indeed S is a feasible solution to problem P.

In the other direction assume that problem P has a feasible solution S. Then, we define an integer flow where the flow along the arc from the root node to the i-th leaf is 1 if $i \in S$ and it is zero otherwise. Then, since $|S| = k$ there are exactly k units of flow leaving the root so the root node has no imbalance. The leaf node i has imbalance $b(i)$ if $i \notin S$ and imbalance $1 - b(i)$ if $i \in S$. Thus, the total imbalance is $\sum_{i\in S}(1 - b(i)) + \sum_{i\notin S} b(i) = \sum_{i\in V} b(i) + \sum_{i\in S} \frac{a_i}{4T} = \frac{1}{4} + \sum_i b(i)$. Its cost is $\sum_{i\in S}(-a_i) = -T$, so indeed this is a YES instance for the decision variant of BI-MCNF. □

3.2. *Polynomial solvable cases of* BI-MCNF

Our goal is to show that if there are only κ fractional values in the vector of node supplies (and the other $|V| - \kappa$ values of node supplies are integer), then there exists an algorithm with time complexity $2^\kappa \cdot T$ where the value of T is upper bounded by a polynomial in $|V|$ and $|E|$. In order to establish this result we will establish the existence of such algorithm for the case $\kappa = 0$, and then show how to use this algorithm together with a guessing step in order to establish the algorithm for general κ.

3.2.1. *The case $\kappa = 0$*

When $\kappa = 0$, which means $b(i)$ is integer for all $i \in V$, the BI-MCNF problem is in fact a standard MCNF problem. Observe that by summing up both hand sides of constraints (3.1b) over $i \in V$, we get $\sum_{i\in V} d_i - \sum_{i\in V} e_i = \sum_{i\in V} b(i)$. Without loss of generality, we can assume that $\sum_{i\in V} b(i) = 0$, in which case $\sum_{i\in V} d_i = \sum_{i\in V} e_i$. Therefore, bounding the total imbalance $\sum_{i\in V}(d_i + e_i)$ by B is equivalent to bounding the total deficit or total excess by $\lfloor B/2 \rfloor$. So the integral BI-MCNF problem on network $G = (V, A)$ is equivalent to the standard MCNF problem on network $G' = (V \cup \{s, t\}, A' \cup \{(t, s)\})$ shown in Fig. 3 where A' results from A by adding two arcs (s, i) and (i, t) of zero cost and infinite capacity for every node $i \in V$. For each $i \in V$, arc (s, i) represents

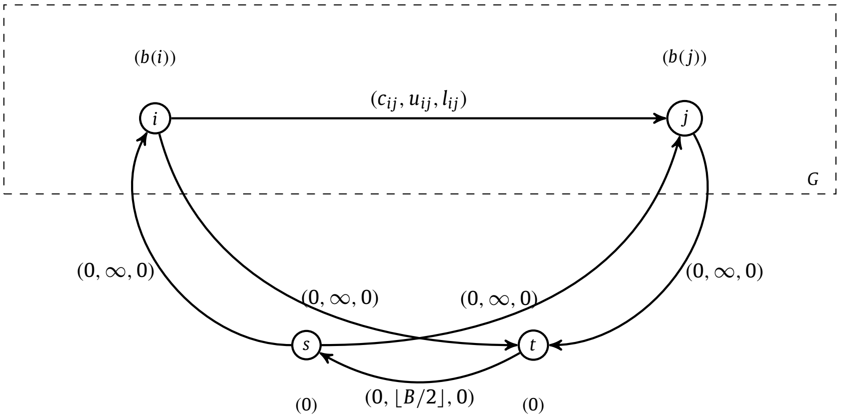

Fig. 3. The schematic transformed standard MCNF network for which an optimal flow solution is optimal for integral BI-MCNF.

variable d_i for each i and arc (i, t) represents variable e_i. Flow on arc (t, s) equals the total deficit and the total excess, so the upper bound capacity of this arc is $\lfloor B/2 \rfloor$. Therefore, by using standard algorithms for the minimum cost network flow problem we can solve the integral BI-MCNF problem in polynomial time.

3.2.2. Reducing the general case to the case of $\kappa = 0$ via guessing

Next, we consider a general instance. We fix an optimal solution OPT. For each node $i \in V$ such that $b(i)$ is fractional (i.e., not integral) we guess if OPT has a strictly positive d_i or strictly positive e_i. In the first case we modify the demand of i to be $\lceil b(i) \rceil$ and decrease the value of B by $1 - \{b(i)\}$ and in the second case we modify the demand of i to be $\lfloor b(i) \rfloor$ and decrease the value of B by $\{b(i)\}$. We apply this guessing for all i without the knowledge of OPT by trying all 2^{κ} possibilities, for each of those possibilities we apply the algorithm we have established for the case where all supplies are integers, and among the feasible instances we have created, we pick the solution whose cost is minimized. The resulting time complexity is 2^{κ} times the time complexity of the algorithm for the special case where all supplies are integers.

The claim that this algorithm returns the optimal solution for BI-MCNF follows by the following observation. First, for every iteration of the exhaustive search resulting in a feasible solution for the resulting instance of BI-MCNF the same assignment to the flow variables corresponds to a feasible solution for the original instance (with the same cost) by increasing the value of d_i (or e_i) by $1 - \{b(i)\}$ (or $\{b(i)\}$, respectively) for nodes with fractional value $b(i)$ whose supply was increased (decreased, respectively) with respect to its value in the original instance.

Second, for the iteration in which we used the new supplies according to OPT the flow variables of OPT results in a feasible solution to the new instance of the same cost. Thus, the optimality of our algorithm follows and we conclude the following result.

Theorem 3.2. *There exists an algorithm for* BI-MCNF *with time complexity $O(2^{\kappa} \cdot T(|V|, |E|))$ where κ is the number of nodes in the instance with non-integral supply, and $T(|V|, |E|)$ is the time complexity for solving the case where the input with $|V|$ nodes and $|E|$ edges satisfies that all supplies are integral.*

Acknowledgement

The research of D.S. Hochbaum and X. Rao is supported in part by NSF award No. CMMI-1760102. A. Levin is supported in part by ISF - Israel Science Foundation grant number 308/18.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, 1993.

[2] P. Amaral, P. Barahona, On optimal correction of inconsistent linear constraints, in: International Conference on Principles and Practice of Constraint Programming, Springer, 2002, pp. 33–46.

[3] J.W. Chinneck, Feasibility and viability, in: Advances in Sensitivity Analysis and Parametric Programming, Springer, 1997, pp. 491–531.

[4] J.W. Chinneck, Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods, vol. 118, Springer Science & Business Media, 2007.

[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, New York, NY, USA, 1978.

[6] D.S. Hochbaum, X. Rao, Network flow methods for the minimum covariates imbalance problem, arXiv preprint, arXiv:2007.06828, 2020.

[7] D.S. Hochbaum, X. Rao, J.J. Sauppe, Network flow methods for the minimum covariate imbalance problem, 2021, submitted for publication.

[8] R.M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, Springer, 1972, pp. 85–103.

[9] J.B. Orlin, A faster strongly polynomial minimum cost flow algorithm, Oper. Res. 41 (1993) 338–350.

[10] G.M. Roodman, Note—post-infeasibility analysis in linear programming, Manag. Sci. 25 (1979) 916–922.

[11] J.J. Sauppe, Balance Optimization Subset Selection: a Framework for Causal Inference with Observational Data, PhD thesis, University of Illinois at Urbana-Champaign, 2015.

[12] J.R. Zubizarreta, Using mixed integer programming for matching in an observational study of kidney failure after surgery, J. Am. Stat. Assoc. 107 (2012) 1360–1371.