

A linear programming approach to the minimum maximum flow degree problem

Manoel Campêlo

Universidade Federal do Ceará
Campus do Pici, Bloco 910, 60455-760 Fortaleza, CE, Brazil
mcampelo@lia.ufc.br

Jhonata Matias

Universidade Federal do Ceará
Campus do Pici, Bloco 910, 60455-760 Fortaleza, CE, Brazil
jhonatadam@lia.ufc.br

RESUMO

Considere uma rede $\mathcal{N}$ com capacidade nos arcos, um nó de destino e um conjunto de nós de origem, onde cada nó de origem tem uma demanda de fluxo inteira positiva a ser enviada através de $\mathcal{N}$ ao nó de destino. Cada possível envio de toda a demanda que não excede a capacidade dos arcos define um fluxo viável. A soma do fluxo entrando e saindo de um nó v de $\mathcal{N}$ em um fluxo viável é o grau de fluxo de v . Dado um fluxo viável, seu grau de fluxo máximo é o maior grau de fluxo em um nó de $\mathcal{N}$. O problema de fluxo de mínimo grau máximo consiste em determinar um fluxo viável com menor grau de fluxo máximo. Introduzimos um modelo de programação inteira mista para o problema e mostramos como resolvê-lo em tempo polinomial. Apresentamos resultados computacionais de experimentos comparando o método proposto com outros dois métodos da literatura.

PALAVRAS CHAVE. Problema de fluxo de mínimo grau máximo, Grau de fluxo, Problema de coloração de fluxo.

Programação inteira mista. Fluxo em rede.

ABSTRACT

Consider a network $\mathcal{N}$ with arc capacities, a sink node and a set of source nodes, where each source node has a positive integer flow demand to be sent through $\mathcal{N}$ to the sink. Each possible sending of the total demand that does not exceed the arc capacities defines a feasible flow. The sum of the flow entering and leaving a node v of $\mathcal{N}$ in a feasible flow is the flow degree of v . Given a feasible flow, its maximum flow degree is the maximum flow degree of the nodes of $\mathcal{N}$. The minimum maximum flow degree problem (MMFDP) consists in determining a feasible flow such that its maximum flow degree is minimum. We introduce a mixed integer linear program for the MMFDP and show how to solve it in polynomial time. We report results of computational experiments that compares the proposed method with two other existing methods.

KEYWORDS. Minimum maximum flow degree problem. Flow degree. Flow coloring problem.

Mixed integer programming. Network flow.

1. Introduction

Let $\mathcal{N} = (V, A, b, c)$ be a network with *node set* V , *arc set* A , *demand function* $b : V \mapsto \mathbb{Z}$ and *capacity function* $c : A \mapsto \mathbb{N} \cup \{\infty\}$. We highlight a special node $t \in V$ to be called *sink node* and consider $b(v) \geq 0, \forall v \in V - t$, and $b(t) = -\sum_{v \in V-t} b(v)$. Let us call $v \in V$ a *source node* if $b(v) > 0$. We use $S(\mathcal{N})$ to denote the set of all source nodes of $\mathcal{N}$ and $b(\mathcal{N})$ to denote $\sum_{v \in S(\mathcal{N})} b(v)$. We let $N^+(v)$ and $N^-(v)$ stand for the set of nodes that have an arc from and to $v \in V$, respectively, and denote by $\Phi(\mathcal{N})$ the set of feasible flows of $\mathcal{N}$, where a *feasible flow* is a function $\phi : A \rightarrow \mathbb{Z}_+$ such that

$$\sum_{u \in N^+(v)} \phi(v, u) - \sum_{u \in N^-(v)} \phi(u, v) = b(v), \quad \forall v \in V, \quad (1)$$

$$\phi(u, v) \leq c(u, v), \quad \forall (u, v) \in A. \quad (2)$$

Furthermore, we call the *flow degree* of a node $v \in V$ in a feasible flow $\phi \in \Phi(\mathcal{N})$ the value

$$d(v, \phi) = \sum_{u \in N^+(v)} \phi(v, u) + \sum_{u \in N^-(v)} \phi(u, v), \quad (3)$$

the *maximum flow degree* in ϕ the value

$$\Delta(\mathcal{N}, \phi) = \max_{v \in V} d(v, \phi),$$

and the *minimum maximum flow degree* of $\mathcal{N}$ the value

$$\Delta_{\Phi}(\mathcal{N}) = \min_{\phi \in \Phi(\mathcal{N})} \Delta(\mathcal{N}, \phi).$$

The *minimum maximum flow degree problem* (MMFDP) consists in finding a feasible flow $\phi^* \in \Phi(\mathcal{N})$ such that $\Delta(\mathcal{N}, \phi^*) = \Delta_{\Phi}(\mathcal{N})$ [Campêlo and Matias, 2021]. Such a feasible flow ϕ^* is said to be *optimal* for the MMFDP on $\mathcal{N}$. We assume that $\Phi(\mathcal{N}) \neq \emptyset$, otherwise the problem is infeasible. Note that feasibility can be checked in polynomial time [Ahuja et al., 1993].

Figure 1(a) presents an MMFDP instance. Notice that vertices v_1 and v_2 have demand 3, vertex t has demand -6 , and the other vertices have null demand. Besides, arcs (v_2, v_5) and (v_5, t) have infinity capacity, arc (v_2, v_4) has null capacity and the other arcs have capacity 2. Figures 1(b) and 1(c) show two feasible flows with maximum flow degrees equal to 8 and 6, respectively, attained at vertex v_5 (and also at vertex t in Figure 1(c)). In particular, the second flow is optimal. Differently from Figure 1(a), in Figures 1(b) and 1(c), the value close to each arc represents the amount of flow that passes through it (arcs that pass no flow are omitted).

The minimum maximum flow degree for uncapacitated networks was initially used in Campêlo et al. [2015] as a theoretical lower bound for the *flow coloring problem* (FCP). In the context of the FCP, $\mathcal{N}$ is a symmetric orientation of an undirected graph G , and the arc capacities are infinite. Each feasible flow $\phi \in \Phi(\mathcal{N})$ defines a multigraph G_{ϕ} where the multiplicity of an edge is given by the flow passing through the corresponding arc. The FCP consists in determining $\min_{\phi \in \Phi(\mathcal{N})} \chi'(G_{\phi})$ [Campêlo et al., 2012], where $\chi'(G_{\phi})$ is the minimum integer $k \geq 0$ such that G_{ϕ} admits a k -edge coloring. The FCP can model bandwidth allocation scenarios under interference constraints in radio networks [Campêlo et al., 2015; Huiban, 2009; Klasing et al., 2008]. The complexity of the FCP for the general graphs remains open, and it is polynomial for bipartite graphs [Campêlo et al., 2015].

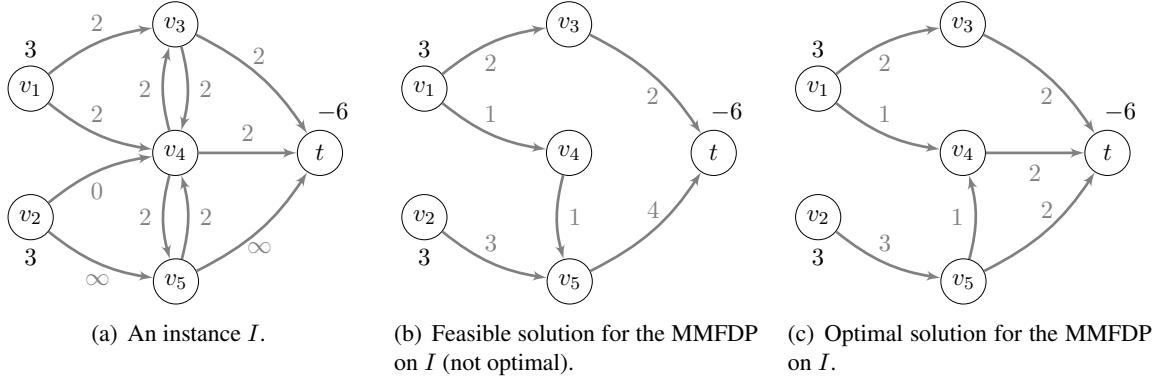


Figure 1: Illustration of the MMFDP.

Other connections between the problems MMFDP and FCP can be seen in Matias and Campêlo [2018]. In that work, $\Delta_\Phi(\mathcal{N})$ is used to derive a theoretical upper bound for the FCP. Besides, it is proposed a heuristic for the FCP that uses an optimal solution of the MMFDP and thus has computational complexity given as a function of the computational complexity of an algorithm for this problem. In Matias and Campêlo [2018], the complexity of MMFDP was not established (even for uncapacitated symmetric networks), and the problem was solved by an integer linear program, introduced in that paper. More recently, Campêlo and Matias [2021] presented an approximation algorithm that, as the heuristic in Matias and Campêlo [2018], builds a solution for the FCP using an optimal solution for the MMFDP. In this second work, a polynomial-time algorithm for the MMFDP is presented. This algorithm performs a binary search between $b(\mathcal{N})$ and $2b(\mathcal{N})$ (lower and upper bounds for the MMFDP, respectively), solving in each step a maximum flow problem over an auxiliary network. For brevity, let us call this algorithm the *binary search approach* (BSA).

Since the MMFDP is a polynomial-time solvable problem, a natural issue is to determine a linear programming formulation for the MMFDP. In this paper, we address this issue. We present a mixed integer linear program that has only two integer variables. Besides, we devise a procedure to find its solution in polynomial time by solving two linear programs. In Section 2, we present the proposed formulation and our procedure to solve it. In addition, we compare this new formulation with the formulation by Matias and Campêlo [2018]. In Section 3, we report computational experiments carried out with the two formulations and the BSA. Finally, we present in Section 4 some conclusions and further research directions.

2. ILP formulations

In this section, we study integer linear programming (ILP) formulations for the MMFDP. We start by introducing a mixed ILP formulation with only two integer variables, and then we compare it with the pure ILP formulation proposed in Matias and Campêlo [2018]. Besides, we show how to solve the new formulation in polynomial time.

2.1. Mixed integer linear programming formulation

The proposed mixed integer linear program for the MMFDP uses variables ϕ_{uv} , for all $(u, v) \in A$, to define a feasible flow (in correspondence with function $\phi : A \rightarrow \mathbb{Z}_+$) as well as one integer variable z and one binary variable w , both related to the $\Delta(\mathcal{N}, \phi)$. Precisely, z and w will respectively give the integer quotient and remainder of the division of $\Delta(\mathcal{N}, \phi)$ by 2. Initially, we consider variables ϕ as integer (we will show later that such integrality constraints are unnecessary). For simplicity, let us call the following model F_1 .

$$\min 2z + w \quad (4)$$

$$\text{s.t.: } \sum_{u \in N^+(v)} \phi_{vu} - \sum_{u \in N^-(v)} \phi_{uv} = b(v), \quad \forall v \in V, \quad (5)$$

$$z + w - \sum_{u \in N^-(v)} \phi_{uv} \geq \left\lceil \frac{b(v)}{2} \right\rceil, \quad \forall v \in V, b(v) \text{ odd}, \quad (6)$$

$$z - \sum_{u \in N^-(v)} \phi_{uv} \geq \frac{b(v)}{2}, \quad \forall v \in V, b(v) \text{ even}, \quad (7)$$

$$0 \leq \phi_{uv} \leq c(u, v), \quad \forall (u, v) \in A, \quad (8)$$

$$\phi \in \mathbb{Z}^{|A|}, \quad (9)$$

$$z \in \mathbb{Z}, w \in \mathbb{B}. \quad (10)$$

Constraints (5) and (8) ensure that ϕ is a feasible flow. Besides, $2z + w$ is at least $\Delta(\mathcal{N}, \phi)$. This is determined by Constraint (6), when $\Delta(\mathcal{N}, \phi)$ is odd, and Constraint (7), when it is even. Finally, the objective function (4) guarantees that $2z + w$ is equal to $\Delta_{\Phi}(\mathcal{N})$ and ϕ is an optimal flow.

The following proposition establishes the correctness of the above formulation.

Proposition 1. *Let (ϕ^*, z^*, w^*) be an optimal solution of F_1 for an MMFDP instance $\mathcal{N} = (V, A, b, c)$. Then $\phi^* \in \Phi(\mathcal{N})$ and $2z^* + w^* = \Delta(\mathcal{N}, \phi^*) = \Delta_{\Phi}(\mathcal{N})$.*

Proof. We start with the following claim.

Claim. Let $\phi \in \Phi(\mathcal{N})$ and $\bar{v} \in V$ such that $d(\bar{v}, \phi) = \Delta(\mathcal{N}, \phi)$. Define $\bar{z} = \sum_{u \in N^-(\bar{v})} \phi_{u\bar{v}} + \lfloor b(\bar{v})/2 \rfloor$ and $\bar{w} = \lceil b(\bar{v})/2 \rceil - \lfloor b(\bar{v})/2 \rfloor$. Then, $(\phi, \bar{z}, \bar{w})$ is feasible for (4)-(10) and $2\bar{z} + \bar{w} = d(\bar{v}, \phi) = \Delta(\mathcal{N}, \phi)$.

Proof of the Claim. By (1) and (3), observe that

$$d(v, \phi) = 2 \sum_{u \in N^-(v)} \phi_{uv} + b(v) \quad \forall v \in V. \quad (11)$$

Then,

$$2\bar{z} + \bar{w} = 2 \sum_{u \in N^-(\bar{v})} \phi_{u\bar{v}} + \lfloor b(\bar{v})/2 \rfloor + \lceil b(\bar{v})/2 \rceil = d(\bar{v}, \phi).$$

Since ϕ is an integer feasible flow and b is an integer function, constraints (5) and (8)-(10) clearly hold. To check constraints (6)-(7), let $v \in V$. If $b(v)$ is odd, then

$$\bar{z} + \bar{w} = \sum_{u \in N^-(v)} \phi_{uv} + \left\lceil \frac{b(v)}{2} \right\rceil = \left\lceil \frac{d(v, \phi)}{2} \right\rceil \geq \left\lceil \frac{d(v, \phi)}{2} \right\rceil = \sum_{u \in N^-(v)} \phi_{uv} + \left\lceil \frac{b(v)}{2} \right\rceil,$$

where the inequality is due to $d(\bar{v}, \phi) \geq d(v, \phi)$. If $b(v)$ is even, then

$$\bar{z} = \sum_{u \in N^-(v)} \phi_{uv} + \left\lfloor \frac{b(v)}{2} \right\rfloor = \left\lfloor \frac{d(v, \phi)}{2} \right\rfloor \geq \frac{d(v, \phi)}{2} = \sum_{u \in N^-(v)} \phi_{uv} + \frac{b(v)}{2},$$

where the inequality is due to $d(\bar{v}, \phi) \geq d(v, \phi)$ and the fact that $d(v, \phi)$ is even (the same parity as $b(v)$). In any case, we conclude that $(\phi, \bar{z}, \bar{w})$ is feasible. $\square$

Let (ϕ^*, z^*, w^*) be an optimal solution of F_1 for a network $\mathcal{N}$. Constraints (5), (8) and (9) define a feasible flow, and so $\phi^* \in \Phi(\mathcal{N})$. Let $\bar{v} \in V$ such that $d(\bar{v}, \phi^*) = \Delta(\mathcal{N}, \phi^*)$. By the claim, we deduce that $2z^* + w^* \leq d(\bar{v}, \phi^*)$. To show the converse inequality, we examine two cases:

- if $b(\bar{v})$ is odd, by Equation (11) and Constraint (6) we have that $d(\bar{v}, \phi^*) \leq 2z^* + 2w^* - 1 \leq 2z^* + w^*$;
- if $b(\bar{v})$ is even, by Equation (11) and Constraint (7) we have that $d(\bar{v}, \phi^*) \leq 2z^* \leq 2z^* + w^*$.

Therefore, $2z^* + w^* = d(\bar{v}, \phi) = \Delta(\mathcal{N}, \phi^*)$.

It remains to show that $\Delta(\mathcal{N}, \phi^*) = \Delta_\Phi(\mathcal{N})$. Suppose not. Since $\phi^* \in \Phi(\mathcal{N})$, it means that there is $\phi' \in \Phi(\mathcal{N})$ such that $\Delta(\mathcal{N}, \phi') < \Delta(\mathcal{N}, \phi^*)$. By the claim, there exists a feasible solution (ϕ', z', w') such that $2z' + w' = \Delta(\mathcal{N}, \phi')$. Then, we have $2z' + w' = \Delta(\mathcal{N}, \phi') < \Delta(\mathcal{N}, \phi^*) = 2z^* + w^*$, which contradicts the fact that $2z^* + w^*$ is the optimal value. $\square$

We now simplify F_1 by removing Constraint (9). Let us call the simplified model by F_2 . To show that F_2 is correct, it suffices to prove the existence of an optimal solution assigning integer values for variables ϕ . This fact is guaranteed by the following property.

Proposition 2. *The constraint matrix related to variables ϕ in F_2 is totally unimodular.*

Proof. Since Constraints (8) define an identity matrix, we only need to consider the matrix R associated with the left hand side of Constraint (5) and Constraints (6)–(7) multiplied by -1 so as to obtain $\leq$ -inequalities. For ease of notation, let us index a row of R associated with $v \in V$ by v' , for (5), and by v'' , for (6) and (7). Let $V' = \{v' : v \in V\}$ and $V'' = \{v'' : v \in V\}$. A column of R associated with arc $(u, v) \in A$ will be simply indexed by uv . Note that column uv of R has only 3 non-null elements: -1 in row v' , $+1$ in row u' and 1 in row v'' . Let I be a subset of indices of the rows of R . We have to partition it into disjoint subsets I_1 and I_2 such that $D_e := \sum_{i \in I_1} r_{ie} - \sum_{i \in I_2} r_{ie} \in \{0, \pm 1\}$, for all $e \in A$ [Wolsey, 1998].

Let $I_1 = (I \cap V') \cup \{v'' \in I \cap V'' : v' \in I\}$ and $I_2 = \{v'' \in I \cap V'' : v' \notin I\}$. Let $e \in A$. Note that $0 \leq \sum_{i \in I_2} r_{ie} \leq \sum_{v'' \in V''} r_{v''e} \leq 1$. First suppose that $\sum_{i \in I_2} r_{ie} = 1 = r_{v''e}$, for some $v'' \in I_2$. Then, $v' \notin I_1$, implying that $\sum_{i \in I_1} r_{ie} \in \{0, 1\}$. Therefore, $D_e \in \{-1, 0\}$. Now, suppose that $\sum_{i \in I_2} r_{ie} = 0$. Let $e = uv$. Then, $D_e = \sum_{i \in I_1} r_{ie}$ and $v'' \notin I_2$. If $v'' \in I_1$, then $v' \in I_1$ and so $D_e = \{0, 1\}$. If $v'' \notin I_1$, then $D_e \in \{0, \pm 1\}$. $\square$

Corollary 1. *Let (z^*, w^*, ϕ^*) be an optimal solution of F_2 , where ϕ^* is a basic solution for F_2 with $z = z^*$ and $w = w^*$. Then $\phi^* \in \Phi(\mathcal{N})$ and $2z^* + w^* = \Delta(\mathcal{N}, \phi^*) = \Delta_\Phi(\mathcal{N})$.*

2.2. Solving F_2

Let us denote by F_2^c the formulation obtained from F_2 by fixing $w = c \in \{0, 1\}$. We present now a method that uses F_2^0 and F_2^1 with a simple cutting-plane algorithm to solve F_2 .

Let (z^c, ϕ^c) be an optimal solution for F_2^c and

$$c' = \arg \min_{c \in \{0, 1\}} 2z^c + c.$$

Notice that $(z^{c'}, c', \phi^{c'})$ is an optimal solution for F_2 , and therefore we can solve F_2 by the following algorithm:

1. for $c \in \{0, 1\}$, find an optimal solution (z^c, ϕ^c) for F_2^c ;

2. if $2z^0 \leq 2z^1 + 1$, then return $(z^0, 0, \phi^0)$;
3. otherwise, return $(z^1, 1, \phi^1)$.

It remains to show how to solve F_2^c . Let $LR(F)$ be the linear relaxation of a formulation F . Consider now (z^c, ϕ^c) as an optimal solution of $RL(F_2^c)$. If $z^c \in \mathbb{Z}$, then (z^c, ϕ^c) is also an optimal solution for F_2^c ; otherwise, by the objective function (4) (which in F_2^c is equivalent to the minimization of z), variable z is least $\lceil z^c \rceil$, that is, inequality $z \geq \lceil z^c \rceil$ is valid for F_2^c . Let (z^*, ϕ^*) be an optimal solution of $RL(F_2^c)$ after including this inequality. We claim that $z^* = \lceil z^c \rceil$ and so (z^*, ϕ^*) is optimal. Otherwise, if $z^* > \lceil z^c \rceil$, there is $\alpha \in (0, 1)$ such that $(1 - \alpha)(z^*, \phi^*) + \alpha(z^c, \phi^c) = (\lceil z^c \rceil, \phi')$ and $(\lceil z^c \rceil, \phi')$ is a feasible solution of $RL(F_2^c)$ (since its feasible set is convex) whose value, $2\lceil z^c \rceil + c$, is strictly less than the optimal value $2z^* + c$: a contradiction.

2.3. Previous integer linear programming formulation

Another (straightforward) ILP formulation for MMFDP was proposed in Matias and Campêlo [2018]. It will be called F_0 and can be expressed as follows:

$$\min z \quad (12)$$

$$\text{s.t.: } \sum_{u \in N^+(v)} \phi_{vu} - \sum_{u \in N^-(v)} \phi_{uv} = b(v), \quad \forall v \in V, \quad (13)$$

$$\sum_{u \in N^+(v)} \phi_{vu} + \sum_{u \in N^-(v)} \phi_{uv} \leq z, \quad \forall v \in V. \quad (14)$$

$$0 \leq \phi_{uv} \leq c(u, v), \quad \forall (u, v) \in A, \quad (15)$$

$$\phi \in \mathbb{Z}^{|A|}, \quad (16)$$

$$z \in \mathbb{Z}. \quad (17)$$

Given a node v of $\mathcal{N}$ and an optimal solution (z^*, ϕ^*) for F_0 on $\mathcal{N}$, the left hand side of inequality (14) associated to v is equal to $d(v, \phi^*)$, according to (3). Consequently, the objective function (12) ensures that $z^* = \Delta_\Phi(\mathcal{N})$.

As originally proposed in Matias and Campêlo [2018], F_0 was restricted to FCP instances, where the capacities of the arcs are unbounded. Here, we have introduced Constraints (15) to address general MMFDP instances.

Differently from F_2 , in F_0 the constraint matrix related to variables ϕ does not have the total unimodularity property. Therefore, we have to keep the integrality constraints for these flow variable. To verify this, consider the square submatrix composed by the intersection of the rows related to a same vertex $v \in V$ in constraints (13) and (14), and columns vu' and $u''v$, where $(v, u'), (u'', v) \in A$. Such a submatrix is

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix},$$

which has determinant 2.

3. Experiments and computational results

Now, we report on the experiments carried out with three solution procedures for the MMFDP. The first one consists in applying the branch-and-cut method to solve F_0 , as proposed in Matias and Campêlo [2018]. In this experiment, we use the branch-and-cut version provided by *CPLEX* in its default setting. The second procedure is the BSA. To solve the maximum flow

problems we use *CPLEX network optimizer*, which is a linear programming solver for problems with network structure. The third one is the method proposed in Subsection 2.2 to solve F_2 . In this latter approach, the linear programs are also solved by *CPLEX network optimizer*.

In the experiments, we mainly evaluate the time performance of the methods. For this purpose, we implement all three methods in C++ with *IBM CPLEX 12.6.0 Concert Technology*, and we perform the experiments in a computer with an *Intel Core i7-3770* processor of 3.40 GHz and *16GB RAM*, running *Ubuntu 14.04 LTS 64 bits*. As test instances, we use the FCP instances generated in Matias and Campêlo [2018], where the arc capacities are unbounded. Starting from them, we also generate two other sets of instances. In the first one, each arc of the original network $\mathcal{N}$ is assigned a capacity, randomly chosen in the interval $[0, b(\mathcal{N})]$. In the second one, we proceed similarly, but now considering the interval $[0, \lceil b(\mathcal{N})/2 \rceil]$. The 540 instances from Matias and Campêlo [2018] are equally divided into 36 groups. Each group, with 15 instances, is generated from one of the possible combinations of the following parameters: number of nodes (150 and 300), arc density (5%, 7% and 10%), percentage of nodes that are source (2%, 4% and 6%) and source demand range (from 1 to 25 and from 1 to 50), where the demand of a source is randomly defined within the range.

In all experiments, we turn off the *CPLEX presolve* in order to keep the ‘flow structure’ of the constraint matrix. This is necessary for *CPLEX network optimizer* to work appropriately. Besides, we observed that this strategy leads to an improvement in the performance of the solution method of F_0 .

Table 1: Experiments with the solution procedures for the MMFDP.

Capacity	N° of nodes	Arc density	F_0	BSA	F_2
∞	150	5%	0.0128	0.0057	0.0027
		7%	0.0172	0.0062	0.0028
		10%	0.0183	0.0070	0.0033
	300	5%	0.0317	0.0131	0.0058
		7%	0.0390	0.0155	0.0075
		10%	0.0563	0.0193	0.0093
$[0, b(\mathcal{N})]$	150	5%	0.0250	0.0058	0.0027
		7%	0.0348	0.0062	0.0030
		10%	0.0422	0.0068	0.0035
	300	5%	0.0863	0.0129	0.0064
		7%	0.1162	0.0154	0.0078
		10%	0.1523	0.0190	0.0100
$[0, \left\lceil \frac{b(\mathcal{N})}{2} \right\rceil]$	150	5%	0.0374	0.0058	0.0028
		7%	0.0514	0.0062	0.0031
		10%	0.0638	0.0069	0.0036
	300	5%	0.1492	0.0131	0.0068
		7%	0.2184	0.0155	0.0078
		10%	0.3106	0.0189	0.0102
All			0.0813	0.0111	0.0055

Table 1 shows the computational results for the performed experiments. Each row corresponds to an instance set with the same arc capacity group (Column 1), number of nodes (Column 2) and arc density (Column 3). The last three columns show the average execution times (in seconds) spent by F_0 , BSA and F_2 , respectively. Each value is the average for 90 instances with the same number of nodes, arc density, and arc capacity range. Finally, the last row shows the average execution time for all instances.

As shown in Table 1, for all tested instance groups, the method related to F_2 presents the best average execution time, followed by the BSA and F_0 . It is worth observing that the performance of the BSA and F_2 does not seem to be affected by the variations in the arc capacities. On the other hand, the computational times of F_0 slightly increase as the capacities become more restrictive, even though the optimal solution was found always in the root node of the branch-and-cut tree.

4. Concluding remarks

In this paper, we proposed a mixed integer linear programming for the MMFDP and presented an algorithm that solves it in polynomial time. We performed experiments comparing this algorithm with two other existing algorithms. In these experiments, the proposed approach presented the best computational times. Besides, it is patent that the elapsed time of this approach is insignificant for the tested instances and grows slowly with their sizes. This points out that the proposed solution procedure could be efficiently embedded in solution methods for the FCP.

In addition to the initial motivation, the study of the MMFDP revealed other attractive issues. We were able to found an ILP formulation with only two integer variables, which can be solved in polynomial time by efficient linear programming algorithms. Once MMFDP is a polynomial-time solvable network flow problem, we can still wonder whether it can be modeled directly as a linear program.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq, Brazil (309315/2019-0), and FUNCAP-CE/Brazil (PNE-0112-00061.01.00/16).

References

- Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network flows: theory, algorithms, and applications*. Prentice Hall.
- Campêlo, M., Corrêa, R., Huiban, C., and Rodrigues, C. D. (2012). The flow coloring problem. In *Congreso Latino-Iberoamericano de Investigacion Operativa (CLAIO)*, p. 3256–3267.
- Campêlo, M., Huiban, C., Rodrigues, C. D., and Sampaio, R. M. (2015). On the complexity of the flow coloring problem. *Discrete Applied Mathematics*, 197:75–92.
- Campêlo, M. and Matias, J. A. (2021). Using the minimum maximum flow degree to approximate the flow coloring problem. *Annals of Operations Research*, p. 1–12.
- Huiban, C. (2009). Radio mesh networks and the round weighting problem. *Université de Nice-Sophia Antipolis (UNS), Sophia Antipolis, France*.
- Klasing, R., Morales, N., and Pérennes, S. (2008). On the complexity of bandwidth allocation in radio networks. *Theoretical Computer Science*, 406(3):225–239.
- Matias, J. and Campêlo, M. (2018). Método exato para o problema de coloração de fluxo. In *Congreso Latino-Iberoamericano de Investigacion Operativa (CLAIO)*, p. 1–8.
- Wolsey, L. A. (1998). Integer programming. series in discrete mathematics and optimization.