Math. Program., Ser. B (2010) 124:271–284
DOI 10.1007/s10107-010-0366-6

FULL LENGTH PAPER

Max Flow and Min Cut with bounded-length paths: complexity, algorithms, and approximation

A. Ridha Mahjoub · S. Thomas McCormick

Received: 3 August 2008 / Accepted: 6 December 2009 / Published online: 11 May 2010

Abstract We consider the "flow on paths" versions of Max Flow and Min Cut when we restrict to paths having at most B arcs, and for versions where we allow fractional solutions or require integral solutions. We show that the continuous versions are polynomial even if B is part of the input, but that the integral versions are polynomial only when $B \leq 3$. However, when $B \leq 3$ we show how to solve the problems using ordinary Max Flow/Min Cut. We also give tight bounds on the integrality gaps between the integral and continuous objective values for both problems, and between the continuous objective values for the bounded-length paths version and the version allowing all paths. We give a primal–dual approximation algorithm for both problems whose approximation ratio attains the integrality gap, thereby showing that it is the best possible primal–dual approximation algorithm.

Mathematics Subject Classification (2000) 90B10 · 90C57 · 68M10

A. Ridha Mahjoub: much of this work was done while at Laboratoire LIMOS, Université Blaise Pascal, Clermont-Ferrand.
S. Thomas McCormick: this research was partially supported by an NSERC Operating Grant, and a Visiting Professorship at Laboratoire LIMOS, Université Blaise Pascal, Clermont-Ferrand; a preliminary version of these results was [31].

A. R. Mahjoub (✉)
Université Paris-Dauphine, LAMSADE, CNRS Place du Maréchal de Lattre de Tassigny,
75775 Paris Cedex 16, France
e-mail: mahjoub@lamsade.dauphine.fr

S. T. McCormick
Sauder School of Business, University of British Columbia, Vancouver, BC V6T 1Z2, Canada
e-mail: Tom.McCormick@sauder.ubc.ca

1 Introduction

It has been well-known for a long time that there are two equivalent formulations of Max Flow and Min Cut problems: The usual formulation has flow variables on arcs, uses the node-arc incidence matrix to express conservation, and has simple upper bounds on flows to express arc capacities. The alternate formulation has flow variables on paths, uses the path-arc incidence matrix to collect path flows into net flows on arcs in order to express arc capacities, and uses non-negativity of the path flows to get conservation. The second formulation is somewhat more natural for seeing that Min Cut is the dual of Max Flow.

Although most people regard the arc flow formulation as being more natural, historically the path flow formulation was developed first, in Ford and Fulkerson's seminal paper proving the Max Flow/Min Cut Theorem [14]. There are many applications where we would like to ensure that no individual unit of flow takes too long to reach its destination. For such applications the path flow formulation is more useful, as we can restrict flows to occur only on paths of bounded length.

A major motivation for this paper is to consider to what extent usual Max Flow/Min Cut results extend from the case where all paths are available to the case where only bounded length paths are available. A second motivation is that Hoffman [21] developed an abstraction of Ford and Fulkerson's path flow model in which a generalized Max Flow/Min Cut Theorem is still true. Unfortunately, Hoffman's proof is non-algorithmic, but Martens and McCormick [28] later developed an algorithm for Hoffman's model based on [30]. Hoffman's model depends on a *crossing axiom* that says, roughly, that when two paths share an arc ("cross"), then there is a third path using the first part of one path and the second part of the other path. Our results show that on both the theory side and the algorithmic side, the crossing axiom is crucial for guaranteeing that there exists an integer-valued Max Flow and Min Cut.

In this paper we show that even with integral data, it is possible to have fractional optimal solutions with arbitrarily large denominators. We completely characterize the integrality gaps between the integral and continuous versions, and give a primal–dual approximation algorithm whose approximation ratio attains these gaps (which is therefore best possible among primal–dual algorithms). Secondly, we note that continuous versions of the problem are polynomial for any bound, but that finding an integer-valued Max Flow or Min Cut in such networks is polynomial only for a bound of at most three. However, in this case we show how to solve the problems using ordinary Max Flow/Min Cut. All of our results apply to both directed and undirected graphs.

Our bounded length Max Flow model is inspired by a real application of Maurras and Vaxès [29] to some routing problem in telecommunications networks, which we will present in the next section. A cut version of the same model arises in work of Pesneau et al. [15] (see also [22–24,32]).

2 Bounded-length path models

2.1 The formal models

In some applications where physical commodities flow through a network, it is necessary that the commodity does not flow through too many arcs on its way to its destination. We can imagine that each arc traversal further degrades the commodity, and then a bound on the number of arcs passed through is a way of specifying a quality level in the delivered product.

This is particularly important in telecommunications networks, where path length is called the number of *hops*. One such application occurs in work of Maurras and Vaxès [29]: For a given source and sink we are interested in how many messages we can route through a network, given that we are only allowed to use paths with at most B hops. A similar scenario arises in distributed authentication systems, where we would like a chain of authentication that is not too long, see e.g., Reiter and Stubblebine [34].

A second application occurs in work of Pesneau et al. [15,22,24,32]: They consider a model for designing a 2-edge-connected network where each edge belongs to a cycle of length at most k. This is to ensure that if any single edge breaks, we can still route traffic between the two ends of the broken edge in at most $k-1$ hops. They give some constraints that model this problem as an integer program. To use these constraints in a Branch and Cut framework, it is necessary to have a separation routine for these constraints. In this case this reduces to asking for a minimum capacity cut hitting all paths with at most $k-1$ hops.

To model this, suppose that we have an ordinary Max Flow network $\mathcal{N} = (N, E)$ with source s, sink t, and capacity u_e on arc e. As usual, let n be the number of nodes of $\mathcal{N}$. Let B be the upper bound on the number of arcs that any unit of flow is allowed to pass through. Define $\mathcal{P}$ to be the set of all s–t directed paths, and $\mathcal{P}(B)$ to be the set of $P \in \mathcal{P}$ with at most B arcs. For each $P \in \mathcal{P}(B)$ let x_P be the flow on path P. Then the problem of *Maximum B-Path Flow* is

$$\begin{aligned}
(\text{Max } B\text{PF}) \qquad \max \sum_{P \in \mathcal{P}(B)} x_P & \\
\text{s.t.} \sum_{P \ni e} x_P \le u_e & \quad \text{for all } e \in E \\
x_P \ge 0 & \quad \text{for all } P \in \mathcal{P}(B).
\end{aligned}$$

If we further require that all x_P be integral, we get the problem *Integral Max BPF*, or Int Max BPF.

Dually, if S is an arc subset such that every $P \in \mathcal{P}(B)$ contains at least one arc of S, then we call S a *B-Path Cut* (BPC). Given capacities u on the arcs, the problem *Integral Min BPC*, or Int Min BPC is to find a BPC S minimizing $u(S)$. The linear relaxation of this is the dual of the Max BPF LP above. The dual LP has a variable y_e on each arc, and is

$$\begin{aligned}
&\text{(Min } B\text{PC)} \quad && \min \sum_e u_e y_e \\
& && \text{s.t.} \sum_{e \in P} y_e \geq 1 \quad \text{for all } P \in \mathcal{P}(B) \\
& && \qquad\quad y_e \geq 0 \quad \text{for all } e \in E.
\end{aligned}$$

Thus Maurras and Vaxès' problem is a case of Max BPF, and Pesneau et al.'s separation problem is a case of Min BPC. In both cases, the value of B that arises in practice tends to be small, somewhere around 3–5. Thus we are particularly interested in cases where B is fixed at a small value.

Note that both these LPs apply equally well to the case where the graph is undirected, just by re-defining $\mathcal{P}(B)$ to include all undirected paths of length at most B. Both applications mentioned above concern undirected graphs.

Notice that as B gets larger, the constraint that flow can occur only on paths of $\mathcal{P}(B)$ gets weaker as $\mathcal{P}(B)$ converges to $\mathcal{P}$. Thus we use Max ∞PF to denote ordinary Max Flow with flow allowed on all paths, and Min ∞PC to denote ordinary Min Cut where all paths of $\mathcal{P}$ must be cut. The Max Flow/Min Cut Theorem says that (for integral data) the objective values of Max ∞PF, Int Max ∞PF, Min ∞PC, and Int Min ∞PC are equal. LP Duality says that the values of Max BPF and Min BPC are equal, and the objective values of the Int versions are (if anything) worse. Since Int Min BPC is less constrained than Int Min ∞PC its objective value may be lower. Thus we get the following relations between the objective values of our problems (where we also mark the pairs of problems where we later analyze integrality gaps):

$$\begin{aligned}
&\text{Min } \infty PC \;=\; \text{Max } \infty PF \;=\; \text{Int Min } \infty PC \;=\; \text{Int Max } \infty PF \\
&\geq \underbrace{\text{Int Min } BPC \;\geq\; \text{Min } BPC}_{\text{int. gap Min } BPC} \;=\; \underbrace{\text{Max } BPF \;\geq\; \text{Int Max } BPF}_{\text{int. gap Max } BPF}.
\end{aligned}$$

2.2 Review of related work

Similar models have been studied by other authors. Many applications arise in telecommunications, where it can be important to limit the number of hops traversed by a message. Thus many papers refer to "hop-constrained" flow instead of length-bounded flow.

In [9] Dahl and Gouveia consider the routing hop-constrained st-path problem. This consists of finding between two distinguished nodes s and t a minimum cost path with no more than B edges when B is fixed. This problem can be solved efficiently using dynamic programming. Dahl and Gouveia describe valid inequalities for the problem and characterize the convex hull of its solutions when $B \leq 3$. Dahl et al. [10] investigate the polytope of the directed st-walks having no more than B arcs, where a directed walk is a directed path that may go through the same node more than once. They present an extended formulation for the underlying B-walk problem when $B = 4$, and use projection to obtain a complete linear description of this polytope.

Coullard et al. [8] study the problem of finding a minimum directed st-walk having exactly B arcs. They present an extended formulation of the problem and, using projection, they give a linear description of the associated polyhedron. Gouveia and Magnanti [17] consider the related problem that consists of finding a minimum spanning tree such that the number of edges between any pair of nodes in the tree is limited to a given bound D (i.e., the diameter). This problem is polynomially solvable if $D \leq 3$ and NP-hard if $D \geq 4$. Gouveia and Magnanti [17] derived single source formulations for the problem based on the concept of tree centers along with some computational experiments. In [18] Gouveia et al. introduce a new modeling approach for the case when D is odd and show that this approach performs better than the one in Gouveia and Magnanti [17].

Kabadi et al. [26] consider the hop-constrained maximum flow and minimum cost flow problems. In particular, they give a strongly polynomial linear programming based algorithm for the continuous versions of the minimum hop-constrained maximum flow problem with arbitrary bound value B. For $B = 3$, they give a strongly polynomial combinatorial algorithm for the integer version of the maximum flow problem.

A very closely related problem, which has received much attention, is the maximum hop-constrained node- (edge-)disjoint paths problem. This consists in finding a maximum number of edge- (resp. node-) s–t paths in $\mathcal{P}(B)$ (Max BEDP) (resp. Max BNDP). This is a special case of our problem with all capacities equal to one. In particular, when all capacities are one, then Int Max BPF models Max BEDP; conversely, if we split an edge of capacity u_e into u_e parallel edges, then Max BEDP models Int Max BPF. Hence Int Max BPF can be no easier than Max BEDF, and it could be harder, since splitting edges into parallel edges could lead to a super-polynomial blowup in size. Lovász et al. [27] were the first to consider this problem. For the Max BNDP, they established a relation analogous to Menger's theorem when $B \leq 4$. For $B \geq 5$, they gave upper and lower bounds for the gap between the maximum number of length-bounded node-disjoint st–paths and the minimum number of edges in a cut. Exoo [12] studies some generalizations of these results to Max BEDP.

Itai et al. [25] study the complexity of several variants of Max BEDP. They show that the problem is NP-complete for $B \geq 5$, which shows that Int Max BPF is also NP Hard for $B = 5$. Itai et al. also show that the problem is polynomially solvable for some of the variants for $B \leq 4$. In particular, they devise a polynomial-time algorithm for Max BNDP (resp. Max BEDP) when $B \leq 4$ (resp. $B \leq 3$) (see [16, Problem ND41]). More complexity results on Max BEDP are in Guruswami et al. [20]. In [20] it is shown that Max BEDP is Max SNP-hard, and that (unless P = NP) it is NP Hard to approximate Max BEDP in the directed case to within a factor of $m^{1/2-\varepsilon}$, and that (unless NP = ZPP) it is NP Hard to approximate Max BEDP in the undirected case to within a factor of $m^{1/2-\varepsilon}$, for any $\varepsilon > 0$. The proofs use very large values of B, which do not correspond well to the values of interest in practice.

In [5,6], Bley addresses approximation and computational issues for the node-disjoint and edge-disjoint hop-constrained path problems. In particular, he shows that the problem of computing the maximum number of edge-disjoint paths between two given nodes of length equal to 3 is polynomial. This result answers an open question in [25]. He also shows that the problem is APX-complete when $B \geq 5$.

In [4], Ben-Ameur defines some classes of 2-connected graphs satisfying path (and cycle)-length constraints. He introduces some parameters and establishes properties and relationships between these graphs. Moreover, he investigates the hop-constrained flow problem and gave lower bounds on the number of edges of these graphs. As a consequence, he obtains some valid inequalities for the underlying survivable network design problem.

A more general model which has been considered in the literature is when each arc or edge of the graph has a length, and then one allows only paths whose total length is at most some bound. Our problem is the special case where all lengths are one. This general model was addressed by e.g., Fleischer and Skutella [13], Baier [2], and Baier et al. [3]. In [13] a fully polynomial time approximation scheme (FPTAS) is given for the continuous length-bound multi-commodity flow problem with non-negative edge lengths. In [2] (see also [3]) Baier considers both the continuous and integral versions of the multi-commodity length-bounded flow problem. For the first version, he gives some structural properties of the optimal solutions and addresses some complexity issues. In particular, he shows that determining (in an outer-planar graph) whether there exists a maximum length-bounded flow between two nodes s and t of a given value is NP-hard. (A graph is outer-planar if it consists of a cycle with non-crossing chords). He also gives an approximation algorithm for solving the maximum length-bounded single commodity flow problem.

For the integral version of the problem, Baier considers in particular the bounded-length single commodity flow problem. He discusses the integrality gap of the integer formulation and its linear relaxation. He also gives some complexity results. In particular, he shows that the problem is NP-hard in simple planar graphs and multi-outer-planars with unit weights and capacities. (Note that the problem in this case is equivalent to find a maximum number of edge-disjoint bounded st-paths.) However for outer-planar simple graphs, it is shown that the problem in this case can be solved by a quasi-polynomial algorithm. Baier [2] also discusses the minimum length-bounded st-cut problem. Some properties of the integrality gap between an integer and a fractional solution are given along with some approximation and complexity results. In particular it is shown that determining whether there exists a fractional length-bounded st-cut of given value is NP-hard even if the graph is outer-planar.

In [3], Baier et al. show that the minimum length-bounded cut problem is NP-hard for $B \geq 4$, and devise approximation algorithms. They consider the maximum length-bounded flow problem and discuss the integrality gap of the linear programming formulation. They also analyze the structure of the optimal solutions.

2.3 Some non-integral examples

We now give a family of counterexamples showing that Max BPF and Min BPC can have optimal solutions with arbitrarily large denominators (a similar result is given by Baier [2, Theorem 2.4]).

Consider the networks $\mathcal{N}_{kp}$ in Fig. 1, which are parametrized by integers $0 < p < k$. The capacities of the k *heavy* arcs are all one, the capacities of all other arcs are infinity. Note that all infinite-capacity arcs occur as two arcs in series to discourage paths that skip too many heavy arcs (using a heavy arc incurs only one arc of path length,

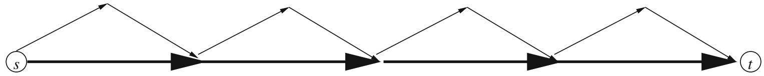

Fig. 1 Counterexample network $\mathcal{N}_{4,2}$ has $B = 6$

whereas skipping incurs two arcs). We choose $B = 2k - p$. This choice of B implies that $\mathcal{P}(B)$ includes exactly the subset of s–t paths which contain at least p heavy arcs. (A version of the network $\mathcal{N}_{3,2}$ was given by Abel et al. [1] to show that max path flow problems do not in general have integral optimal solutions; $\mathcal{N}_{3,2}$ was given by Itai et al. [25, Fig. 10] to show that Menger's theorem does not hold for bounded-length path problems, an early version of the problems that concern us here.)

Note that $\mathcal{N}_{kp}$ has $\binom{k}{p}$ paths containing exactly p heavy arcs. A fixed heavy arc is contained in $\binom{k-1}{p-1}$ of these paths. Using these observations, it is easy to see that one optimal solution to Max BPF for $\mathcal{N}_{kp}$ puts flow $1/\binom{k-1}{p-1}$ on each of the $\binom{k}{p}$ paths containing exactly p heavy arcs, for a total flow value of k/p. One way to see this is to note that the unique min BPC puts a value of $1/p$ on each heavy arc, and zero elsewhere (for a dual objective value of k/p), and the primal and dual solutions are complementary slack. LP theory shows that Max BPF for $\mathcal{N}_{kp}$ always has an optimal solution with at most k positive paths, so other optimal solutions exist. However, for $\mathcal{N}_{k,k-1}$ the given solution to Max BPF (value $1/(k-1)$ on each of the k paths with $k-1$ heavy arcs) *is* unique, so both Max BPF and Min BPC can have fractional solutions with arbitrarily large denominators.

Note that for all $\mathcal{N}_{kp}$, the optimal objective values of Max ∞PF and Min ∞PC value are ∞, for all k and p. Thus this family of examples shows that the ratio between the Max ∞PF and Min ∞PC objective value and the Max BPF and Min BPC objective value can be arbitrarily large.

It is also easy to see that the optimal value of Int Max BPF is $\lfloor k/p \rfloor$, and for Int Min BPC is $k - p + 1$ for all $\mathcal{N}_{kp}$. It is natural to wonder how large the ratio between the optimal values of Int Max BPF and Max BPF, and Int Min BPC and Min BPC can be, the so-called *integrality gap*.

For this class of examples, the integrality gap for Max BPF is $\frac{k/p}{\lfloor k/p \rfloor}$. This is maximized when k is odd and $p = (k + 1)/2$, since the numerator is asymptotically 2, and the denominator equals 1, for an asymptotic ratio of 2.

For this class of examples, the integrality gap for Min BPC is $\frac{k-p+1}{k/p}$. By calculus this is maximized again when k is odd and $p = (k+1)/2$. In this case the numerator is asymptotically $k/2$, and the denominator is asymptotically 2, for an asymptotic ratio of $k/4$, or $B/6 = \Theta(B)$. Boyles and Exoo [7] appear to have been first to construct examples with integrality gaps of $\Theta(B)$ for Min BPC, in fact for Min BEDP. Baier [2, Lemma 2.22] (see also [3, Theorem 13]) constructed more complicated examples (based on a construction in [20]) that yield integrality gaps also of size $\Theta(B)$.

We now develop an approximation algorithm for Int Min BPC and Int Max BPF with a ratio of $O(B)$ by applying the primal–dual framework [35, Chapters 8–10]. The algorithm works by constructing a(n integer) feasible path flow x to the LP (Max BPF) and a BPC C which satisfies the part of complementary slackness that says that $e \in C$ implies that $\sum_{P \ni e} x_P = u_e$ (while ignoring the other parts of complementary slackness):

Primal–Dual Algorithm for Int Min BPC and Int Max BPF

Set $x \leftarrow 0$, $C \leftarrow \emptyset$
While C is not a BPC
    Find a path P of length $\leq B$ not cut by C
    Increase x_P until some arc $e \in P$ satisfies $\sum_{Q \ni e} x_Q = u_e$
    Set $C \leftarrow C \cup \{e\}$
Return C

Theorem 2.1 *This algorithm is a B-approximation algorithm for Int Min BPC and Int Max BPF. Furthermore, the integrality gap between Int Min BPC and Min BPC is $\Theta(B)$, and between Int Max BPF and Max BPF is also $\Theta(B)$, implying that this approximation algorithm is best possible primal–dual approximation algorithm for this formulation.*

Proof Clearly this algorithm returns a feasible Int BPC C, and it runs in polynomial time (each trip through the loop costs $O(m)$ time to find if a violating P exists, and we can make at most m trips through the loop since at least one arc is added to C each time).

To analyze the performance of the algorithm, note that, because of partial complementary slackness,

$$\sum_{e \in C} u_e = \sum_{e \in C} \sum_{P \ni e} x_P = \sum_{P \in \mathcal{P}(B)} |P \cap C| x_P \leq B \sum_{P \in \mathcal{P}(B)} x_P,$$

since each P gets counted once for each $e \in C$ on P, and $|P \cap C| \leq |P| \leq B$. Define Z to be the joint optimal value of (Max BPF) and (Min BPC). Then since the incidence vector of C is feasible to (Min BPC) we get

$$Z \leq \sum_{e \in C} u_e \leq B \sum_{P \in \mathcal{P}(B)} x_P \leq B \cdot Z,$$

proving that this algorithm has an approximation ratio of B. Note that the algorithm also produces a feasible solution x_P to Int Max BPF with an approximation ratio of B, by the same proof.

As noted above, the integrality gap between Int Min BPC and Min BPC for $\mathcal{N}_{kp}$ with $B = 2k - p$ with $k \geq 1$ odd and $p = (k+1)/2$ is $\Theta(B)$. Baier et al. [3, Theorem 12] (following an example of [20]) give a family of examples of Int Max BPF whose integrality gaps with Max BPF is also $\Theta(B)$. As is well-known [35], approximation ratios coming from the primal–dual algorithm can never be better than the integrality gap. Since here we have families of examples with integrality gaps of $\Theta(B)$, this algorithm is the best we can do without either relying on a tighter formulation of Int Min BPC or Int Max BPF, or by using a different type of approximation algorithm. □

As mentioned in Sect. 2.2, the approximation results of [20] for the Max BEDP use impractically large values of B. The simple $O(B)$ approximation algorithm here is likely to be more useful in practice.

Note that $\mathcal{N}_{3,2}$ has $B = 4$. The Int Max 4PF value of $\mathcal{N}_{3,2}$ is 1, but the Int Min 4PC value is 2. This shows that the Max Flow/Min Cut result for $B = 3$ of Sect. 3.2 has the largest value of B where this is still true. (This was essentially already pointed out in [25, Fig. 10].)

Since the set of paths $\mathcal{P}(B)$ for the examples in this section is the same whether the graphs are directed or undirected, and since the approximation algorithm works the same for undirected graphs, all the results in this section apply equally well to undirected graphs.

3 The complexity of Max BPF and Min BPC

3.1 The continuous versions are polynomial

We first consider the complexity of the continuous versions. If B is fixed (i.e., not part of the input), then the cardinality of $\mathcal{P}(B)$ is $O(n^B)$, which is polynomial. Then Max BPF and Min BPC are just polynomial-size linear programs, which can be solved in polynomial time.

Suppose now that B is *not* fixed. The Separation Problem for Min BPC has a vector $\hat{y}$ as input, and asks for the shortest path in $\mathcal{P}(B)$ w.r.t. lengths $\hat{y}$. This length-restricted shortest path problem can be solved in polynomial time via dynamic programming. By the equivalence between Separation and Optimization for dual LPs [19], this shows that Max BPF and Min BPC are again polynomial even when B is not fixed. This shortest path of bounded length subproblem is solved as easily for undirected as for directed graphs, hence the results here apply also to the undirected case.

When B is not fixed, Max BPF and Min BPC are not *compact* LPs, in that Max BPF can have an exponential number of variables, and Min BPC can have an exponential number of constraints. Adapting an idea of Martin et al. [11] we can get equivalent compact LPs for both problems containing only polynomial numbers of variables and constraints.

Define an expanded network $\mathcal{N}'$ containing $B+1$ copies of the nodes of $\mathcal{N}$ denoted $N^0, N^1, \ldots, N^B$. The copy of $i \in N$ in N^k is called i^k. For each arc $i \to j$ of $\mathcal{N}$ and each $0 \le k < B$ make an arc $i^k \to j^{k+1}$ with (variable) length y_{ij}. For each $i \in N$ and each $0 \le k < B$ make an arc $i^k \to i^{k+1}$ with (fixed, constant) length 0. Then there is an obvious 1–1 mapping from s–t paths $P \in \mathcal{P}(B)$ in $\mathcal{N}$, and paths from s^0 to t^B in $\mathcal{N}'$. Therefore y is a feasible solution to Min BPC iff the minimum value of potential difference $\pi_{t^B} - \pi_{s^0}$ in $\mathcal{N}'$ is at least 1. Thus we can write Min BPC as the compact LP:

$$
\begin{aligned}
&(\text{Min } B\text{PC}) \qquad \min \sum_e u_e y_e \\
&\text{s.t. } \pi_{i^k} - \pi_{j^{k+1}} \ge y_{ij} \quad \text{for all arcs } i \to j \text{ of } \mathcal{N} \text{ and all } k \\
&\pi_{i^k} - \pi_{i^{k+1}} \ge 0 \quad \text{for all nodes } i \in N \text{ and all } k \\
&\pi_{s^0} - \pi_{t^B} \ge 1 \\
&y_e \ge 0 \quad \text{for all } e \in E.
\end{aligned}
$$

The dual of this LP is a compact formulation of Max BPF. It has an arc flow variable z_{ij}^k on each arc $i^k \to j^{k+1}$ of $\mathcal{N}'$, and a constraint on the sum of the z_{ij}^k over k that ensures that any decomposition of arc flow z into a flow on paths in $\mathcal{P}(B)$ must satisfy the capacity constraint:

$$
\begin{aligned}
\text{(Max } B\text{PF)} \quad \max \quad & \sum_{j \in N: s \to j \in E} z_{sj}^0 \\
\text{s.t.} \quad & \sum_{i \in N: i \to l \in E} z_{il}^k = \sum_{j \in N: l \to j \in E} z_{lj}^{k+1} \quad \text{for all nodes } l^k \neq s^0,\ t^B \text{ in } \mathcal{N}' \\
& \sum_{k=0}^{B-1} z_{ij}^k \leq u_{ij} \quad \text{for all } i \to j \in E \\
& z_{ij}^k \geq 0 \quad k = 0, \ldots, B, i \to j \in E.
\end{aligned}
$$

Here sj, il, and lj represent arcs $s \to j$, $i \to l$, and $l \to j$ in E, respectively. Again the same idea works as well for undirected as for directed networks. These LPs could now be solved by any LP solver without having to use the Ellipsoid Algorithm.

Theorem 3.1 *Max BPF and Min BPC have compact linear programming formulations, and so can be solved in polynomial time, even when B is part of the input.*

3.2 Complexity of Int Max BPF and Int Min BPC

We now switch to considering the complexity of the integral versions of the problems. As mentioned in Sect. 2.2 Itai et al. [25] showed that Int Max BPF is Strongly NP Hard even for $B \geq 5$, and Baier et al. [3, Theorem 4] showed that Int Min BPC is Strongly NP Hard even for $B \geq 4$. Alternate proofs appear in [31].

We now consider small values of B. Although Int Max BPF was already shown to be polynomial for $B \leq 3$ by [25] using essentially the same construction we use here, we re-prove it here in a way that shows the same for Int Min BPC.

Theorem 3.2 *Int Max BPF and Int Min BPC can be solved in polynomial time when $B \leq 3$ for both directed and undirected graphs.*

Proof Given such an undirected instance $\mathcal{N} = (N, E)$ of Int Max 3PF or Int Min 3PC, we construct a directed instance $\tilde{\mathcal{N}} = (\tilde{N}, \tilde{E})$ of Max ∞PF/Min ∞PC as follows (see Fig. 2 for an example): First note that any s to t edges are essentially independent of the rest of the problem, and so can be dealt with separately, so we can assume that no s — t edges exist. Second, any node i which does not have an edge s — i or i — t cannot belong to any path of at most 3 edges and so can be deleted. Thus if we define $S = \{i \in N \mid s — i \in E\}$ and $T = \{j \in N \mid j — t \in E\}$, then $S \cup T = N$.

Let T' be a disjoint copy of T (where we denote the copy of $j \in T$ that is in T' by j'), and set $\tilde{N} = \{s\} \cup \{t\} \cup S \cup T'$. For each edge $s — i \in E$ with capacity u_{si} make arc $s \to i \in \tilde{E}$ also with capacity u_{si}, for each $j — t \in E$ make $j' \to t \in \tilde{E}$ with capacity u_{jt}, and for each $i — j \in E$ with $i, j \notin \{s, t\}$, if $i \in S$ and $j \in T$

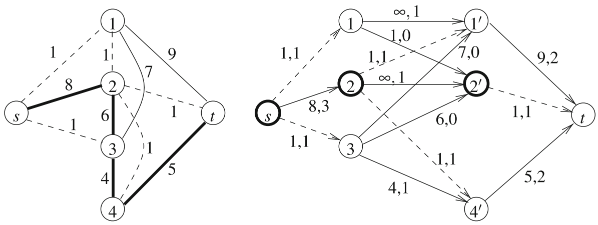

Fig. 2 Example of the construction of Theorem 3.2, with $\mathcal{N}$ on the left with u_e on each edge e, $\tilde{\mathcal{N}}$ on the right with u_{ij}, x^*_{ij} on each arc $i \to j$, where x^* is an ordinary max flow. The *dashed* edges of $\mathcal{N}$ are a Min 3PC of capacity 5 (note that they do not cut the heavy path of four arcs), which corresponds to the dashed arcs of the ordinary min cut induced by the three bold nodes of $\tilde{\mathcal{N}}$

make arc $i \to j'$ with capacity u_{ij} (thus if $i, j \in S \cap T$ and $i — j \in E$, then $i \to j', j \to i' \in \tilde{E}$). For each $i \in S \cap T$ make an arc $i \to i' \in \tilde{E}$ with capacity infinity. Note that there is a 1–1 correspondence between the arcs from S to T' in $\tilde{\mathcal{N}}$ and the paths of length at most 3 in $\mathcal{N}$. That is, path $s — i — j — t$ in $\mathcal{N}$ corresponds to arc $i \to j'$, and path $s — i — t$ corresponds to arc $i \to i'$.

We claim that there is a correspondence between feasible 3PFs x in $\mathcal{N}$ and feasible flows $\tilde{x}$ in $\tilde{\mathcal{N}}$ that preserves objective value, and a correspondence between minimal feasible 3PCs in $\mathcal{N}$ and minimal finite capacity cuts in $\tilde{\mathcal{N}}$ that also preserves objective value.

Suppose that x is a feasible 3PF in $\mathcal{N}$. If we have paths $P_1 = s — i — j — t$ and $P_2 = s — j — i — t$ with $x_{P_1} > 0$ and $x_{P_2} > 0$, then we can modify x as follows: Let $\delta = \min(x_{P_1}, x_{P_2})$, and define $P_3 = s — i — t$ and $P_4 = s — j — t$. Now put $x_{P_1} \leftarrow x_{P_1} - \delta$, $x_{P_2} \leftarrow x_{P_2} - \delta$, $x_{P_3} \leftarrow x_{P_3} + \delta$, and $x_{P_4} \leftarrow x_{P_4} + \delta$. This new x is still feasible, and has the same objective value, and one of x_{P_1} or x_{P_2} has become zero. So for each $i — j \in E$ with $i, j \in S \cap T$, we can assume that at least one of x_{sijt} and x_{sjit} is zero.

For $i \to j' \in \tilde{E}$ corresponding to path P, define $\tilde{x}_{ij'}$ to be x_P. For $i \to j$ equal to $s \to j$ or $i \to t$ in $\tilde{E}$, define $x_{ij} = \sum\{x_P \mid i — j \text{ is an edge of } P\}$. By the previous paragraph this is a feasible flow in $\tilde{\mathcal{N}}$ with the same objective value.

Conversely, given flow $\tilde{x}$ in $\tilde{\mathcal{N}}$, for $i — j \in E$ with $i, j \in S \cap T$ we can use the same trick to modify $\tilde{x}$ to ensure that at most one of $\tilde{x}_{ij'}$ and $\tilde{x}_{ji'}$ is positive. Then put x_P equal to $\tilde{x}_{ij'}$ for the arc $i \to j'$ corresponding to P. This flow x is feasible, and has the same value as $\tilde{x}$.

Suppose that D is a minimal feasible 3PC in $\mathcal{N}$. If $s — i$ or $j — t$ is in D, add arc $s \to i$ or $j \to t$ to $\tilde{D}$. We say that $i — j$ is *gone* if either $i — j \in D$ or $i — j \notin E$. If $i — j \in D$ with $i, j \notin \{s, t\}$, then at least one of $s — i$, $i — t$ must be gone, and at least one of $s — j$, $j — t$ must be gone. If $s — i$ and $s — j$ were both gone, then we could drop $i — j$ from D, contradicting minimality of D; similarly, we could not have both $i — t$ and $j — t$ gone. Thus either the pair $\{s — i, j — t\}$ is gone, or the pair $\{s — j, i — t\}$ is gone, but not both. If both $s — i$ and $j — t$ are gone, then add $j \to i'$ to $\tilde{D}$, else add $i \to j'$ to $\tilde{D}$. Then $\tilde{D}$ has the same capacity as D, and it is easy to check that it is a minimal finite capacity cut in $\tilde{\mathcal{N}}$.

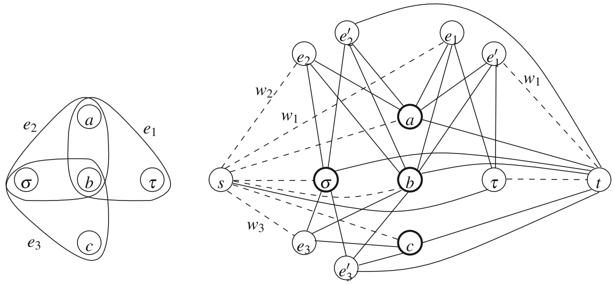

Fig. 3 Example of the construction of Corollary 3.3, with the instance of MHC on the left, and the constructed instance of Min 3PC on the right. If $w_1 < w_2 + w_3$, then the optimal objective value of MHC is w_1, coming from the optimal partition $S = \{\sigma, a, b, c\}, T = \{\tau\}$. The nodes of S are bolded on the right, and the dashed edges are the edges of C. Edge weights are shown only for the s — e_i and e_i' — t in C to reduce clutter. Notice that, e.g., t — $e_2' \notin C$ because every possible path t — e_2' — v — s with $v \in e_2$ already has v — $s \in C$. On the other hand, e.g., s — e_1 and s — e_1' must both be in C to cut the paths s — e_1 — a — t and t — e_1' — τ — s, respectively. The optimal objective value of 3PC is $5M + (M + w_1)$, where the $5M$ term comes from the five s — v and v — t pairs, and the $M + w_1$ term comes from the s — e_i and e_i' — t in C

Conversely, suppose that $\tilde{D}$ is a minimal finite capacity cut in $\tilde{\mathcal{N}}$, so that $\tilde{D}$ does not contain any $i \to i'$ arc. This means that for each $i \in N$, at least one of $s \to i$ and $i' \to t$ is gone. By minimality of $\tilde{D}$, $i \to j' \in \tilde{D}$ only if neither of $s \to i$ and $j' \to t$ is gone. This implies that if $i \to j' \in \tilde{D}$, then $j \to i' \notin \tilde{D}$. Thus we can define D to contain each s — i with $s \to i \in \tilde{D}$, each j — t with $j' \to t \in \tilde{D}$, and each i — j such that one of $i \to j'$ or $j' \to i$ is in $\tilde{D}$, and D will be a minimal 3PC in $\mathcal{N}$ with the same capacity as $\tilde{D}$.

This proof works even more simply in the case where $\mathcal{N}$ is directed. This shows that we can compute Int Max 3PFs and Int Max 3PCs using just one call to Max Flow/Min Cut, and that the Max Flow/Min Cut Theorem holds between Int Max 3PF and Int Max 3PC. □

This proof gives a polynomial algorithm for what looks at first like an unrelated problem: Min Hypergraph Cut (MHC). We are given a hypergraph H on vertex set V with two distinguished vertices σ and τ, and with edge set E (i.e., each $e \in E$ is a subset of V). Each edge $e \in E$ has weight $w_e \geq 0$. The question is to find a partition (cut) $V = S \cup T$ of V with $\sigma \in S, \tau \in T$, such that its weight $w(S, T) = \sum\{w_e \mid e \cap S \neq \emptyset,\ e \cap T \neq \emptyset\}$ is minimum.

Corollary 3.3 *Min Hypergraph Cut can be solved in polynomial time.*

Proof To reduce an instance of MHC to an instance of Int Min 3PC, make two disjoint copies E and E' of E, and put $N = \{s\} \cup \{t\} \cup V \cup E \cup E'$ (see Fig. 3). Choose $M = 1 + 2\sum_{e \in E} w_e$. Connect s to τ with an edge of weight ∞, and to every other $v \in V$ by an edge of weight $M < \infty$, and to every $e \in E$ by an edge of weight w_e. Similarly connect t to σ with an edge of weight ∞, and to every other $v \in V$ by an edge of weight $M < \infty$, and to every $e' \in E'$ by an edge of weight w_e. For each $e \in E$, connect e to each $v \in V$ such that $v \in e$ with an edge of weight ∞, and for each $e' \in E'$, connect e' to each $v \in V$ such that $v \in e$ with an edge of weight ∞.

Now solve Int Min 3PC on this graph, getting a cut C. Clearly C must include exactly one of the edges $s — v$ and $v — t$ for each $v \in V$. Thus if we define $S = \{v \in V \mid s — v \in C\}$ and $T = \{v \in V \mid v — t \in C\}$, then S and T partition V, and we have $\sigma \in S$ and $\tau \in T$, so (S, T) is a hypergraph cut. Conversely, any hypergraph cut (S, T) yields such a partition, and the contribution of these edges to $w(C)$ is $M|V|$, independent of (S, T).

If $e \in E$ has $e \subseteq S$ then C must contain $s — e$ (to cut paths $s — e — v — t$ with $v \in e$) but not $e' — t$ (since all paths $t — e' — v — s$ with $v \in e$ already have $v — s \in C$); similarly, if $e \in E$ has $e \subseteq T$ then C must contain $e' — t$ but not $s — e$; otherwise (i.e., e contributes w_e to $w(S, T)$) then C must contain both $s — e$ and $e' — t$. Hence if e contributes 0 to $w(S, T)$ then it contributes w_e to $w(C)$, and if e contributes w_e to $w(S, T)$ then it contributes $2w_e$ to $w(C)$, so the (S, T) induced from C must solve Min Hypergraph Cut. □

Acknowledgments We thank an anonymous referee for detailed and helpful comments, Jean-François Maurras for bringing the motivating example to our attention, Michel Goemans for pointing out the application of [11] to this problem, Bruce Shepherd for stimulating conversations, and participants at the 2002 Aussois Combinatorial Optimization Workshop for many helpful comments.

References

1. Abel, U., Carstens, H.G., Deuber, W., Prömel, H.J.: Time Dependent and Hypergraphic Networks. Technical Report, Fakultät für Mathematik, Universität Bielefeld. Part of this report appeared as: “On Hypergraphic Networks” in Operations Research Verfahren, **32**(1979), 1–4 (1978)
2. Baier, G.: Flows with Path Restrictions. Ph.D. thesis, TU Berlin (2003)
3. Baier, G., Erlebach, T., Hall, A., Kolman, P., Köhler, E., Pangrac, O., Schilling, H., Skutella, M.: Length-bounded cuts and flows. To appear in Transactions on Algorithms; an extended abstract appears in Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP’06), LNCS 4051, Springer, Berlin, pp. 679–690 (2006)
4. Ben-Ameur, W.: Constrained length connectivity and survivable networks. Networks **36**, 17–33 (2000)
5. Bley, A.: Node-Disjoint Length-Restricted Paths. Master’s thesis, TU Berlin (1997)
6. Bley, A.: On the complexity of vertex-disjoint length-restricted path problems. Comput. Complex. **12**, 131–149 (2003)
7. Boyles, S.M., Exoo, G.: A counterexample to a conjecture on paths of bounded length. J. Graph Theory **6**, 205–209 (1982)
8. Coullard, C.R., Gamble, A.B., Liu, J.: The *K*-walk polyhedron. In: Du, D.-Z., Sun, J. (eds.) Advances in Optimization and Approximation, pp. 9–29. Kluwer, Norwell (1994)
9. Dahl, G., Gouveia, L.: On the directed hop-constrained shortest path problem. Oper. Res. Lett. **32**, 15–22 (2004)
10. Dahl, G., Foldnes, N., Gouveia, L.: A note on hop-constrained walk polytopes. Oper. Res. Lett. **32**, 345–349 (2004)
11. Martin, R.K., Rardin, R.L., Campbell, B.A.: Polyhedral characterization of discrete dynamic programming. Oper. Res. **38**, 127–138 (1990)
12. Exoo, G.: On line disjoint paths of bounded length. Discret. Math. **44**, 317–318 (1983)
13. Fleischer, L., Skutella, M.: Quickest flows over time. SIAM J. Comput. **36**, 1600–1630; an extended abstract appeared in IPCO 2002, LNCS **2337**, Springer, pp. 36–53 (2007)
14. Ford, L.R. Jr., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399–404 (1956)
15. Fortz, B., Mahjoub, A.R., McCormick, S.T., Pesneau, P.: The 2-edge connected subgraph problem with bounded rings. Math. Prog. **105**, 85–111 (2006)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)

17. Gouveia, L., Magnanti, T.L.: Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. Networks **41**, 159–173 (2003)
18. Gouveia, L., Magnanti, T.L., Requejo, C.: A 2-path approach for odd-diameter-constrained minimum spanning and Steiner trees. Networks **44**, 254–265 (2004)
19. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988)
20. Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B., Yannakakis, M.: (2003) Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. J. Comput. Syst. Sci. **67**, 473–496; a preliminary version appeared in Proceedings of 31st STOC, pp. 19–28 (1999)
21. Hoffman, A.J.: A generalization of max flow- min cut. Math. Prog. **6**, 352–359 (1974)
22. Huygens, D., Labbé, M., Mahjoub, A.R., Pesneau, P.: The two-edge hop-constrained network design problem: valid inequalities and branch-and-cut. Networks **49**, 116–133 (2007)
23. Huygens, D., Mahjoub, A.R.: Integer programming formulations for the two 4-hop-constrained paths problem. Networks **49**, 135–144 (2007)
24. Huygens, D., Mahjoub, A.R., Pesneau, P.: Two edge hop-constrained paths and polyhedra. SIAM J. Discret. Math. **18**, 287–312 (2004)
25. Itai, A., Perl, Y., Shiloach, Y.: The complexity of finding maximum disjoint paths with length constraints. Networks **12**, 277–286 (1982)
26. Kabadi, S.N., Kang, J., Chandrasekaran, R., Nair, K.P.K.: Hop Constrained Network Flows: Analysis and Synthesis. Technical report, University of New Brunswick (2004)
27. Lovász, L., Neumann-Lara, V., Plummer, M.: Mengerian theorems for paths of bounded length. Periodica Mathematica Hungarica **9**, 269–276 (1978)
28. Martens, M., McCormick, S.T.: A Polynomial Algorithm for Weighted Abstract Flow. Working paper, Sauder School of Business, UBC; an extended abstract appears in Proceedings of IPCO 2008, pp. 97–111 (2008)
29. Maurras, J-F., Vaxès, Y.: Multicommodity netwok flow with jump constraints. Discret. Math. **165/166**, 481–486 (1997)
30. McCormick, S.T.: A Polynomial Algorithm for Abstract Maximum Flow. UBC Faculty of Commerce Working Paper 95-MSC-001. An extended abstract appears in Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 490–497 (1995)
31. McCormick, S.T.: The Complexity of Max Flow and Min Cut with Bounded Length Paths. Talk at 6th International Workshop on Combinatorial Optimization, Aussois (2002)
32. Pesneau, P.: Conception de Réseaux 2-Connexes avec Contraintes de Bornes. Ph.D. thesis, Université Blaise Pascal, Clermont-Ferrand (in French) (2003)
33. Prömel, H.J.: On Hypergraphic Networks. Universität Bielefeld, Diplomarbeit (1978)
34. Reiter, M.K., Stubblebine, S.G.: Path Independence for Authentication in Large-Scale Systems. AT&T Labs technical report (1996)
35. Williamson, D.P.: Lecture Notes on Approximation Algorithms. IBM TJ Watson Research Center Technical Report RC 21409, Yorktown Heights (1998)