

A flow model based on polylinking system

Michel X. Goemans · Satoru Iwata ·
Rico Zenklusen

Received: 14 March 2010 / Accepted: 1 February 2011
© Springer and Mathematical Optimization Society 2011

Abstract We introduce polylinking networks, which is a flow model based on polylinking systems that generalizes the classical flow model of Ford and Fulkerson on acyclic networks and has applications in the context of wireless networks. More precisely, a flow model recently introduced by Avestimehr et al. (Proceedings of the Allerton conference on communication, control, and computing 2007) used in the context of wireless information networks is a special case of the presented model. We define a notion of source-destination cut and derive a max-flow min-cut theorem. Additionally, we present various properties of polylinking networks that can be seen as generalizations of properties for classical flows. Using submodular function minimization and submodular flow algorithms, one can efficiently determine a maximum flow, a minimum source-destination cut, as well as a minimum cost flow. These algorithms lead to new efficient algorithms for the information flow model.

An earlier shorter version of this article [9] focused mainly on the relation between the ADT model and linking flows.

M. X. Goemans was supported by NSF contract CCF-0829878 and by ONR grant N00014-05-1-0148.

R. Zenklusen was supported by the Swiss National Science Foundation, grant number: PBEZP2-129524.

M. X. Goemans

Department of Mathematics, M.I.T., Room 2-351, Cambridge, MA 20139, USA
e-mail: goemans@math.mit.edu

S. Iwata

Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502, Japan
e-mail: iwata@kurims.kyoto-u.ac.jp

R. Zenklusen (✉)

Department of Mathematics, M.I.T., Room 2-332, Cambridge, MA 20139, USA
e-mail: ricoz@math.mit.edu

Keywords Matroid · Linking system · Wireless information flow · Submodular function

Mathematics Subject Classification (2000) 90C27 · 90C90

1 Introduction

In this work, we introduce a flow model based on combined polylinking systems, a notion introduced by Schrijver [13]. Given is a set of vertices that can be partitioned into layers V_1, \dots, V_r and flow is sent across these layers from V_1 to V_2 and so on. There is no notion of edges, and how flow can be sent from one layer to the next one is described by polylinking systems. Thanks to the abstract nature of polylinking systems, the described model is very general and gives large freedom in specifying how flow can be sent through the network. In particular, the new model includes the classical flow model of Ford and Fulkerson restricted to acyclic graphs as well as a wireless information flow model introduced by Avestimehr et al. [2].

We define a natural notion of source-destination cut which can be seen as a generalization of the notions used in the flow models mentioned above. Using results from polylinking systems we show that the value of a maximum flow is equal to the value of a minimum source-destination cut. Despite the generality of the presented flow model, using submodular flow algorithms, one can find a maximum flow, a minimum source-destination cut as well as a minimum cost flow in polynomial time if an oracle for evaluating a function associated with the polylinking systems used in the description of the network is available.

We give a description of the flow polytope which is the set of all flows. Unlike the classical Ford and Fulkerson flow model, the flow polytope of the presented flow model has in general an exponential number of facets. However, using the reduction to the submodular flow polyhedron, we show that the flow polytope is integral if the underlying polylinking systems are integral. This generalizes the integrality property of classical flows.

In the unit-capacity case the notion of polylinking systems reduces to the notion of linking systems and our algorithmic approach reduces to matroid partition. For this special case, our flow model corresponds to a *cascading system* introduced by Schrijver [13], who presented a polynomial time algorithm for finding maximum flows and minimum cost flows for this setting. The algorithm of Schrijver works by finding augmenting paths in a network which is a union of basis exchange graphs of matroids induced by the linking systems. It is essentially a particular method for solving the above-mentioned matroid partition problem.

The matroid partition approach provides an efficient algorithm for finding a maximum flow and a minimum cut in the ADT flow model. This algorithm is faster than the combinatorial algorithm recently presented by Amaudruz and Fragouli [1]. Furthermore, introducing the notion of aggregation of polylinking systems, we present a more compact formulation of the ADT model. With the aid of an effective use of FFT, this formulation leads to another efficient method of analyzing the ADT model.

The paper is organized as follows. In Sect. 2, we recall some properties of polylinking systems shown by Schrijver [13]. In Sect. 3, we introduce a new flow model and derive several properties, including the max-flow min-cut theorem and integrality of the flow polytope. In Sect. 4, we show that a flow model recently introduced by Avestimehr, Diggavi and Tse [2] for information flows in wireless networks is a special case of our model. In Sect. 5, we present a matroid partition algorithm for the unit-capacity case and discuss its consequences in the ADT model. In Sect. 6, we introduce the notion of aggregation of polylinking system to derive a compact representation of the ADT model as a polylinking network. An efficient method of realizing the function evaluation oracle is also developed. Finally, we conclude this paper in Sect. 7.

2 Polylinking systems and polylinking functions

For $k \in \mathbb{N}$ we use the notation $[k] = \{1, \dots, k\}$. For $x, y \in \mathbb{R}^k$, we denote by $x \vee y \in \mathbb{R}^k$ the component-wise maximum of x and y and by $|x|$ denote the L_1 -norm of x , which is also called *the size* of x . We write $x \leq y$ if the inequality holds component-wise and $x < y$ if $x \leq y$ and $x \neq y$. For a finite set V and $Y \subseteq V$, we denote by $\chi_Y \in \{0, 1\}^V$ the incidence vector of Y . Let $x \in \mathbb{R}^A$, and we denote by $x(Y)$ the sum of the components of x corresponding to Y , i.e., $x(Y) = \sum_{v \in Y} x(v)$. We say that x lies in Y if $x(v) = 0$ for all $v \in V \setminus Y$.

2.1 Polylinking systems

Polylinking systems were introduced by Schrijver [13]. In the following we recall some of their properties. For proofs and more information, see [13].

Definition 1 A *polylinking system* is a triple (V_1, V_2, L) , where V_1 and V_2 are finite sets and L is a non-empty compact subset of $\mathbb{R}_+^{V_1} \times \mathbb{R}_+^{V_2}$ satisfying the following (P1)–(P4).

- (P1) If $(x_1, x_2) \in L$, then $|x_1| = |x_2|$.
- (P2) If $(x_1, x_2) \in L$ and $0 \leq y_1 \leq x_1$, then $\exists y_2 \leq x_2$ with $(y_1, y_2) \in L$.
- (P3) If $(x_1, x_2) \in L$ and $0 \leq y_2 \leq x_2$, then $\exists y_1 \leq x_1$ with $(y_1, y_2) \in L$.
- (P4) If $(x_1, x_2), (y_1, y_2) \in L$, then $\exists (z_1, z_2) \in L$ such that $x_1 \leq z_1 \leq x_1 \vee y_1$ and $y_2 \leq z_2 \leq x_2 \vee y_2$.

For a polylinking system (V_1, V_2, L) , we call V_1 and V_2 the ground sets of the polylinking system. For $(x_1, x_2) \in L$ we say that x_1 is linked to x_2 and vice versa. Schrijver [13, Theorem 6.3] showed that L is a polytope. We denote by (V_2, V_1, \bar{L}) the polylinking system defined by

$$\bar{L} = \{(x_2, x_1) \mid (x_1, x_2) \in L\}.$$

If L is an integral polytope, we say that the polylinking system (V_1, V_2, L) is *integral*. If (V_1, V_2, L) is integral and $L \subseteq [0, 1]^{V_1} \times [0, 1]^{V_2}$, then the polylinking system is simply called a *linking system* [14], which is equivalent to a *bimatroid* introduced

by Kung [10]. Since for a linking system the vertices of L can be seen as incidence vectors of the finite ground sets V_1, V_2 , linking systems are often presented using a set notation instead of a notation relying on incidence vectors. Using set notations, Definition 1 can be rewritten for linking systems as follows. A triple (V_1, V_2, Λ) , where $\emptyset \neq \Lambda \subseteq 2^{V_1} \times 2^{V_2}$, is a linking system if it satisfies the following properties.

- (L1) If $(P_1, P_2) \in \Lambda$, then $|P_1| = |P_2|$.
- (L2) If $(P_1, P_2) \in \Lambda$ and $Q_1 \subseteq P_1$, then $\exists Q_2 \subseteq P_2$ with $(Q_1, Q_2) \in \Lambda$.
- (L3) If $(P_1, P_2) \in \Lambda$ and $Q_2 \subseteq P_2$, then $\exists Q_1 \subseteq P_1$ with $(Q_1, Q_2) \in \Lambda$.
- (L4) If $(P_1, P_2), (Q_1, Q_2) \in \Lambda$, then $\exists (R_1, R_2) \in \Lambda$ such that $P_1 \subseteq R_1 \subseteq P_1 \cup Q_1$ and $Q_2 \subseteq R_2 \subseteq P_2 \cup Q_2$.

In the following we give three examples of polylinking systems, where the first two examples are linking systems. For proofs and further information, see [13].

Example 1 (Linking systems induced by bipartite graphs, also called deltoid linking systems) Let $G = (V_1, V_2, E)$ be an undirected bipartite graph and let $\Lambda \subseteq 2^{V_1} \times 2^{V_2}$ be the set of all pairs (P_1, P_2) of $P_1 \subseteq V_1$ and $P_2 \subseteq V_2$ such that there exists a matching M in G with $\partial M \cap V_1 = P_1$ and $\partial M \cap V_2 = P_2$, where ∂M denotes the set of end-vertices of the edges in M . Then (V_1, V_2, Λ) forms a linking system.

Example 2 (Linking systems induced by matrices, also called representable linking systems) Let K be a field and $A \in K^{V_1 \times V_2}$ be a matrix over the field K , where V_1 and V_2 are two finite sets representing the rows and columns of A . Let $\Lambda \subseteq 2^{V_1} \times 2^{V_2}$ be the set of all pairs (P_1, P_2) of $P_1 \subseteq V_1$ and $P_2 \subseteq V_2$ such that the submatrix of A determined by the row set P_1 and the column set P_2 is a nonsingular matrix. Then (V_1, V_2, Λ) forms a linking system.

Example 3 (Polylinking systems induced by networks, also called gammoid polylinking systems) Let $G = (V, A)$ be a directed graph with capacities $c : A \rightarrow \mathbb{R}_+$ on its edges. Let $V_1, V_2 \subseteq V$ and let $L \subseteq \mathbb{R}_+^{V_1} \times \mathbb{R}_+^{V_2}$ be defined as follows: $(x_1, x_2) \in L$ if and only if there is a feasible flow in G such that the demand given by x_2 can be satisfied by the supply given by x_1 . Then (V_1, V_2, L) is a polylinking system. If G is a bipartite graph with bipartition $V_1 \cup V_2 = V$, then the obtained polylinking system is called a *deltoid polylinking system*.

2.2 Polylinking functions

The *polylinking function* $\lambda : 2^{V_1} \times 2^{V_2} \rightarrow \mathbb{R}_+$ of a polylinking system (V_1, V_2, L) is defined by

$$\lambda(P_1, P_2) = \max\{x_1(P_1) - x_2(V_2 \setminus P_2) \mid (x_1, x_2) \in L\}.$$

For our purposes the following equivalent definition, which follows from the monotonicity property of polylinking systems, seems often more intuitive:

$$\lambda(P_1, P_2) = \max\{|x_1| \mid (x_1, x_2) \in L, x_1(V_1 \setminus P_1) = x_2(V_2 \setminus P_2) = 0\}.$$

Thus, $\lambda(P_1, P_2)$ is the maximum size of a vector lying in P_1 that can be linked to a vector lying in P_2 . A polylinking system is integral if and only if its polylinking function is integral. In the case of linking systems we use the term *linking function* for the polylinking function. Polylinking functions completely describe the underlying polylinking systems as highlighted by the following proposition.

Proposition 1 ([13, Theorem 6.3]) *Let (V_1, V_2, L) be a polylinking system with polylinking function λ . Let $x_1 \in \mathbb{R}_+^{V_1}, x_2 \in \mathbb{R}_+^{V_2}$. Then $(x_1, x_2) \in L$ if and only if $|x_1| = |x_2|$ and $x_1(P_1) - x_2(V_2 \setminus P_2) \leq \lambda(P_1, P_2)$ for all $P_1 \subseteq V_1$ and $P_2 \subseteq V_2$.*

The following proposition gives a characterization of polylinking functions.

Proposition 2 ([13, Theorem 6.3]) *Let V_1, V_2 be two finite sets and let $\lambda : 2^{V_1} \times 2^{V_2} \rightarrow \mathbb{R}_+$. Then λ corresponds to a polylinking system on ground sets V_1 and V_2 if and only if the following conditions hold for all $P_1, Q_1 \subseteq V_1$ and $P_2, Q_2 \subseteq V_2$:*

- (F1) $\lambda(\emptyset, V_2) = \lambda(V_1, \emptyset) = 0$,
- (F2) if $Q_1 \subseteq P_1$ and $Q_2 \subseteq P_2$ then $\lambda(Q_1, Q_2) \leq \lambda(P_1, P_2)$,
- (F3) $\lambda(P_1 \cap Q_1, P_2 \cup Q_2) + \lambda(P_1 \cup Q_1, P_2 \cap Q_2) \leq \lambda(P_1, P_2) + \lambda(Q_1, Q_2)$.

The third condition (F3) in the above proposition is called bisubmodularity. Let V_1, V_2 be two finite sets and let $\lambda : 2^{V_1} \times 2^{V_2} \rightarrow \mathbb{R}_+$ be a function satisfying the conditions of Proposition 2. Then the polylinking system (V_1, V_2, L) corresponding to λ is given by

$$L = \left\{ (x_1, x_2) \in \mathbb{R}_+^{V_1} \times \mathbb{R}_+^{V_2} \mid |x_1| = |x_2|, x_1(P_1) - x_2(V_2 \setminus P_2) \leq \lambda(P_1, P_2) \right. \\ \left. \forall P_1 \subseteq V_1, P_2 \subseteq V_2 \right\}.$$

Let V be a finite set and $\rho : 2^V \rightarrow \mathbb{R}$ be a monotone, submodular function with $\rho(\emptyset) = 0$. Then

$$P(\rho) = \{x \in \mathbb{R}_+^V \mid x(P) \leq \rho(P), \forall P \subseteq V\}$$

is a *polymatroid* on V associated with ρ . There are many relations between polylinking systems and polymatroids. Below we highlight one of them.

Proposition 3 ([13, Theorem 6.3]) *Let (V_1, V_2, L) be a polylinking system and $P_1 \subseteq V_1$. Then the set*

$$\left\{ x_2 \in \mathbb{R}_+^{V_2} \mid \exists x_1 \in \mathbb{R}_+^{V_1} \text{ with } (x_1, x_2) \in L, x_1(V_1 \setminus P_1) = 0 \right\}$$

forms a polymatroid on V_2 .

For a submodular function $\rho : 2^V \rightarrow \mathbb{R}_+$ with $\rho(\emptyset) = 0$, which is not necessarily monotone, the base polyhedron $B(\rho)$ is defined by

$$B(\rho) = \left\{ x \in \mathbb{R}_+^V \mid x(P) \leq \rho(P), \forall P \subseteq V, x(V) = \rho(V) \right\}.$$

The following two propositions show that a polylinking system can be seen as a particular way of representing a base polyhedra and vice versa. This tight link between polylinking systems and base polyhedra allows for deducing many results for polylinking systems from results for base polyhedra.

Proposition 4 ([13, Theorem 6.3]) *Let (V_1, V_2, L) be a polylinking system with linking function λ . Consider the set function $\rho : 2^V \rightarrow \mathbb{R}$ on $V = V_1 \cup V_2$ defined by*

$$\rho(P) = \lambda(V_1 \cap P, V_2 \setminus P).$$

Then ρ is a nonnegative submodular function with $\rho(\emptyset) = \rho(V_2) = \rho(V) = 0$. Moreover, $\rho(P) = 0$ if $P \subseteq V_2$ or $P \supseteq V_2$. The base polyhedron $B(\rho)$ satisfies $B(\rho) = \{(x_1, -x_2) \mid (x_1, x_2) \in L\}$.

In particular for linking systems, we have the following results.

Proposition 5 ([14, Theorem 3.2]) *Let (V_1, V_2, Λ) be a linking system with two disjoint ground sets. Then the set $\mathcal{B}_\Lambda = \{P_1 \cup (V_2 \setminus P_2) \mid (P_1, P_2) \in \Lambda\}$ forms the set of bases of a matroid on the ground set $V_1 \cup V_2$.*

For a linking system (V_1, V_2, Λ) with disjoint ground sets we denote by M_Λ the matroid corresponding to the bases \mathcal{B}_Λ as defined in Proposition 5, and by ρ_Λ the rank function of M_Λ . The following proposition shows the relation between the rank function ρ_Λ and the linking function λ .

Proposition 6 ([14, Theorem 3.2]) *Let (V_1, V_2, Λ) be a linking system with disjoint ground sets and with linking function λ . For $P_1 \subseteq V_1$ and $P_2 \subseteq V_2$,*

$$\rho_\Lambda(P_1 \cup P_2) = \lambda(P_1, V_2 \setminus P_2) + |P_2|.$$

2.3 Combining polylinking systems and polymatroids

The following proposition describes a natural way of how two polylinking systems that share one of the ground sets can be combined together yielding a new polylinking system. This operation is also called the *product* of two polylinking systems.

Proposition 7 ([13, Theorem 6.7]) *Let (V_1, V_2, L_1) and (V_2, V_3, L_2) be two polylinking systems, with polylinking functions λ_1 and λ_2 and define*

$$L_1 \star L_2 = \{(x_1, x_3) \in \mathbb{R}_+^{V_1} \times \mathbb{R}_+^{V_3} \mid \exists x_2 \in \mathbb{R}_+^{V_2} \text{ with } (x_1, x_2) \in L_1, (x_2, x_3) \in L_2\}.$$

Then $(V_1, V_3, L_1 \star L_2)$ is again a polylinking system with polylinking function

$$(\lambda_1 \star \lambda_2)(P_1, P_3) = \min_{P_2 \subseteq V_2} \{\lambda_1(P_1, P_2) + \lambda_2(V_2 \setminus P_2, P_3)\}.$$

Notice that the product of two linking systems is again a linking system. A polylinking system can also be combined with a polymatroid that is defined on one of its ground sets to obtain another polymatroid. This operation is also called the *product* between a polymatroid and a polylinking system.

Proposition 8 ([13, Theorem 6.4]) *Let (V_1, V_2, L) be a polylinking system with polylinking function λ and $P(\rho)$ be a polymatroid with ground set V_1 and rank function ρ . Define $P(\rho) \star L$ by*

$$P(\rho) \star L = \{x_2 \in \mathbb{R}_+^{V_2} \mid \exists x_1 \in P(\rho) \text{ with } (x_1, x_2) \in L\}.$$

Then $P(\rho) \star L$ is a polymatroid with rank function

$$(\rho \star \lambda)(P_2) = \min_{P_1 \subseteq V_1} \{\rho(V_1 \setminus P_1) + \lambda(P_1, P_2)\}.$$

Notice that the product of a matroid with a linking system is again a matroid.

3 A flow model based on polylinking systems

3.1 Polylinking networks

Let $r \geq 2$ be an integer, V_1, \dots, V_r be finite disjoint sets, and (V_i, V_{i+1}, L_i) be a polylinking system with polylinking function λ_i for each $i \in [r - 1]$. The tuple $G = (V, L)$, where $V = (V_1, \dots, V_r)$ and $L = (L_1, \dots, L_{r-1})$ is called a *polylinking network*. The sets V_1, \dots, V_r are called the *layers* of the network G and elements of these sets are called vertices. Furthermore, V_1 is called the *source layer* and its vertices are called *sources*. Similarly, V_r is called the *destination layer* and its vertices are called *destinations* or *sinks*. If $G = (V, L)$ is a polylinking network such that its underlying polylinking systems are linking systems, we say that G is a *linking network*. A *flow in a polylinking network*, or also simply called *flow*, is a tuple $x = (x_1, \dots, x_r) \in \mathbb{R}_+^{V_1} \times \dots \times \mathbb{R}_+^{V_r}$ such that $(x_i, x_{i+1}) \in L_i$ for $i \in [r - 1]$. The value of a flow x is defined by $\phi(x) = |x_1|$. Notice that by definition of a flow we have $|x_1| = |x_i|$ for $i \in [r]$. We say that a flow x goes through some vertex $v \in V_i$ if $x_i(v) > 0$. If there is no danger of ambiguity we sometimes represent a flow as a vector in $\mathbb{R}_+^{\cup_{i \in [r]} V_i}$. A V_1 - V_r *cut in a polylinking network*, also called *source-destination cut* or simply *cut*, is a tuple $C = (C_1, \dots, C_r)$ such that $C_i \subseteq V_i$ for $i \in [r]$, $C_1 = V_1$ and $C_r = \emptyset$. The value of a cut C is defined by

$$\kappa(C) = \sum_{i=1}^{r-1} \lambda_i(C_i, V_{i+1} \setminus C_{i+1}).$$

If there is no danger of ambiguity we also represent a cut $C = (C_1, \dots, C_r)$ by the set $\cup_{i=1}^r C_i$ and use the cut function κ also for this cut representation.

A slightly more general version of the proposed flow model can be obtained by defining a flow to be a tuple $x = (x_1, \dots, x_r)$ that satisfies the polylinking conditions mentioned above and furthermore satisfies that x_1 is contained in some given polymatroid $P(\rho)$. However, this slightly generalized model can easily be cast to the polylinking flow model by adding an additional layer at the beginning to the polylinking system as follows. We add a new set $V_0 = \{v_0\}$ containing only one vertex and define the corresponding polylinking system (V_0, V_1, L_0) by

$$L_0 = \{(x_0, x_1) \in \mathbb{R}_+ \times \mathbb{R}_+^{V_1} \mid x_0 = |x_1|, x_1 \in P(\rho)\}.$$

It is easy to check that the above defined system is a polylinking system and that within the layers 1 to r , flows in the extended network correspond to flows in the initial network with the additional restriction on the first layer given by the polymatroid $P(\rho)$. The above construction also shows that we could without loss of generality restrict ourselves to polylinking networks with a single source. An identical construction could also be applied at the last layer to get a single sink.

Example 4 (Polylinking flows induced by deltoid polylinking systems) Let $G = (V, L)$ be a polylinking network with r layers, where every polylinking system (V_i, V_{i+1}, L_i) is induced by a bipartite graph $(V_i \cup V_{i+1}, E_i)$ with capacities w_i on E_i . Polylinking flows in G correspond to flows in the network obtained from the network $(\cup_{i=1}^r V_i, \cup_{i=1}^{r-1} E_i)$ with capacities $\{w_i\}_{i \in [r-1]}$ by orienting all edges towards the layers with higher indices. Furthermore, the notion of cut in the polylinking network corresponds to the cut notion in classical flow networks.

The above example also shows that classical flows in acyclic graphs can be seen as a special case of polylinking flows since every classical acyclic flow network can be transformed into a layered network by splitting arcs.

3.2 Max-flow min-cut duality and submodularity

By combining Propositions 3 and 7, we get the following gammoid-like property for polylinking network.

Proposition 9 *Let $G = (V, L)$ be a polylinking network with r layers. The set $\{x_r \in \mathbb{R}_+^{V_r} \mid \exists \text{ flow } (x_1, \dots, x_r) \text{ in } G\}$ forms a polymatroid.*

One of the main properties of polylinking networks is the following duality between maximum flows and minimum cuts.

Theorem 1 *In any polylinking network, the value of a maximum flow is equal to the value of a minimum cut.*

Proof Let $G = (V, L)$ be a polylinking network with r layers and we define $\tilde{L} = L_1 \star \dots \star L_{r-1}$. By the definition of the product of polylinking systems we have that there is a linking flow in G of some given value $q \in \mathbb{R}_+$ if and only if there exists a tuple $(x_1, x_r) \in \tilde{L}$ with $|x_1| = q$. The maximum possible value of $|x_1|$ for a linked pair

$(x_1, x_r) \in \tilde{L}$ is given by $\tilde{\lambda}(V_1, V_r)$, where $\tilde{\lambda}$ is the polylinking function corresponding to \tilde{L} (see the equivalent definition of the polylinking function given at the beginning of Sect. 2.2). Thus, the value of a maximum polylinking flow in G is $\tilde{\lambda}(V_1, V_r)$. By Proposition 7 and the definition of the value of a cut we have

$$\begin{aligned} \tilde{\lambda}(V_1, V_r) &= \min \left\{ \lambda_1(V_1, V_2 \setminus P_2) + \sum_{i=2}^{r-2} \lambda_i(P_i, V_{i+1} \setminus P_{i+1}) \right. \\ &\quad \left. + \lambda_r(P_{r-1}, V_r) \mid P_2 \subseteq V_2, \dots, P_{r-1} \subseteq V_{r-1} \right\} \\ &= \min \left\{ \kappa \left(V_1 \cup \bigcup_{i=2}^{r-1} P_i \right) \mid P_2 \subseteq V_2, \dots, P_{r-1} \subseteq V_{r-1} \right\}, \end{aligned}$$

which is the value of a minimum cut in G . □

Theorem 2 *Let $G = (V, L)$ be a polylinking network over r layers, $C = (C_1, \dots, C_r)$ a cut in G and $x = (x_1, \dots, x_r)$ a flow in G . Then C is a minimum cut and x is a maximum flow if and only if*

$$\lambda_i(C_i, V_{i+1} \setminus C_{i+1}) = x_i(C_i) - x_{i+1}(C_{i+1}), \quad \forall i \in [r - 1].$$

Proof Since x is a flow, we have by Proposition 1

$$\lambda_i(C_i, V_{i+1} \setminus C_{i+1}) \geq x_i(C_i) - x_{i+1}(C_{i+1}), \quad \forall i \in [r - 1].$$

Thus, the value of the cut C can be bounded as follows.

$$\begin{aligned} \kappa(C) &= \sum_{i=1}^{r-1} \lambda_i(C_i, V_{i+1} \setminus C_{i+1}) \geq \sum_{i=1}^{r-1} [x_i(C_i) - x_{i+1}(C_{i+1})] \\ &= x_1(C_1) - x_r(C_r) = x_1(V_1) - x_r(\emptyset) = |x_1|. \end{aligned} \tag{1}$$

The bound given by (1) is a way of deriving weak duality, i.e., the value of any flow is smaller or equal than the value of any cut. By the strong duality given by Theorem 1 we have that C is minimum and x is maximum if and only if the inequality in (1) is tight, which proves the theorem. □

Theorem 3 *The function κ is submodular.*

Proof Let $G = (V, L)$ be a polylinking network with r layers let $C = (C_1, \dots, C_r)$ be a cut in G . By definition the value of the cut C is given by $\kappa(C) = \sum_{i=1}^{r-1} \lambda_i(C_i, V_{i+1} \setminus C_{i+1})$. By the bisubmodularity property of polylinking functions, we have for $i \in [r - 1]$ that the function g_i defined on $2^{V_i \cup V_{i+1}}$ by $g_i(P_i \cup P_{i+1}) = \lambda_i(P_i, V_{i+1} \setminus P_{i+1})$, where $P_i \subseteq V_i, P_{i+1} \subseteq V_{i+1}$, is submodular; this property follows also immediately from Proposition 4. Hence, $\kappa(C)$ is the sum of submodular functions, and thus submodular. □

Thus, if for some polylinking network the underlying polylinking functions can be evaluated in strongly polynomial time, then a minimum cut can be determined in strongly polynomial time by minimizing the submodular cut function.

In the case of a linking network, an efficient algorithm for finding a minimum cut can easily be transformed into an efficient algorithm for finding a maximum flow as follows. Consider the elements in $\cup_{i=1}^r V_i$ in any order. If removing an element from the corresponding linking system does not decrease the value of a minimum cut, we remove the element from the graph. One can easily check that the incidence vector of the remaining elements form a maximum flow.

3.3 Optimization in polylinking networks

In this section, we present an efficient method for finding a maximum flow and a minimum cut in a polylinking network. The method is based on reduction to the submodular flow problem introduced by Edmonds and Giles [7]. The presented algorithm runs in strongly polynomial time if the polylinking functions underlying the given polylinking network can be evaluated in strongly polynomial time.

Let $D = (W, A)$ be a directed graph with vertex set W and arc set A . For each node v , we denote the set of arcs leaving v and those entering v by δ^+v and δ^-v , respectively. The boundary $\partial x : W \rightarrow \mathbb{R}$ of a function $x : A \rightarrow \mathbb{R}$ is defined by

$$\partial x(v) = \sum_{a \in \delta^+v} x(a) - \sum_{a \in \delta^-v} x(a).$$

Suppose we have lower and upper bounds $\underline{c} : A \rightarrow \mathbb{R} \cup \{-\infty\}$ and $\bar{c} : A \rightarrow \mathbb{R} \cup \{+\infty\}$, and a submodular function $f : 2^W \rightarrow \mathbb{R}$ with $f(\emptyset) = f(W) = 0$. A function $x : A \rightarrow \mathbb{R}$ is a *submodular flow* if it satisfies $\partial x \in \mathbf{B}(f)$ and $\underline{c}(a) \leq x(a) \leq \bar{c}(a)$ for all $a \in A$. The *submodular flow polyhedron* $\text{SFP}(D, f)$ is the set of submodular flows in \mathbb{R}^A . If \underline{c} , \bar{c} , and f are integral, then the submodular flow polyhedron is known to be integral.

Consider a polylinking network $G = (V, L)$ with $V = (V_1, \dots, V_r)$ and $L = (L_1, \dots, L_{r-1})$. A flow $x = (x_1, \dots, x_r)$ satisfies $(x_i, x_{i+1}) \in L_i$ for each $i = 1, \dots, r - 1$. We now reduce the problem of finding a maximum flow in G to a submodular flow problem.

Let V^+ and V^- be two copies of V . For any subset $S \subseteq V$, we denote its copies in V^+ and V^- by S^+ and S^- , respectively. Construct a directed network $(V^+ \cup V^-, A)$ with vertex set $V^+ \cup V^-$ and arc set $A = V$, where the tail and head of the arc $v \in A$ are the copies $v^+ \in V^+$ and $v^- \in V^-$, respectively. Then shrink the vertices in V_1^- to a single vertex s and the vertices in V_r^+ to a single vertex t . Add an arc $a^* = (s, t)$ to A . In the resulting digraph $D = (W, A)$ with $A = V \cup \{a^*\}$, set the lower and upper bounds to be 0 and $+\infty$, respectively, for all arcs. Consider a submodular function $f : 2^W \rightarrow \mathbb{R}_+$ defined by

$$f(U) = \sum_{i=1}^{r-1} \lambda_i (V_i \cap P, V_{i+1} \setminus Q), \tag{2}$$

where $P \subseteq V$ is the set of all vertices that correspond to vertices in $U \cap V^+$, i.e., $P^+ = U \cap V^+$, and similarly $Q \subseteq V$ consists of all vertices that correspond to vertices in $U \cap V^-$, i.e., $Q^- = U \cap V^-$. Note that $f(\emptyset) = f(W) = 0$. Then we have the following characterization.

Theorem 4 *A vector $x = (x_1, x_2, \dots, x_r) \in \mathbb{R}^V$ is a polylinking flow in G if and only if $x^* = (x_1, x_2, \dots, x_r, \phi(x)) \in \mathbb{R}^A$ is a submodular flow in D .*

Proof Let $x = (x_1, x_2, \dots, x_r)$ be a polylinking flow in G . By Proposition 1 we have for $P, Q \subseteq V$

$$x_i(V_i \cap P) - x_{i+1}(V_{i+1} \cap Q) \leq \lambda(V_i \cap P, V_{i+1} \setminus Q)$$

and $|x_i| = |x_{i+1}|$ for $i \in [r - 1]$. For any $P \subseteq V \setminus V_r$ and $Q \subseteq V \setminus V_1$, we have

$$\begin{aligned} \partial x(P^+ \cup Q^-) &= \sum_{i=1}^{r-1} [x_i(V_i \cap P) - x_{i+1}(V_{i+1} \cap Q)] \\ &\leq \sum_{i=1}^{r-1} \lambda_i(V_i \cap P, V_{i+1} \setminus Q) = f(P^+ \cup Q^-). \end{aligned}$$

Since $\partial x(s) = \partial x(t) = 0$, this implies $\partial x(U) \leq f(U)$ for any $U \subseteq W$ with $U \cap V^+ = P^+$ and $U \cap V^- = Q^-$. Thus x is a submodular flow in D .

Conversely, let $x' = (x_1, \dots, x_r, x_\infty)$ be a submodular flow in D . Then

$$\partial x'(V_i^+ \cup V_{i+1}^-) \leq f(V_i^+ \cup V_{i+1}^-) = 0$$

holds for $i \in [r - 1]$. In addition, $\partial x'(s) \leq f(\{s\}) = 0$ and $\partial x'(t) \leq f(\{t\}) = 0$. Since $\partial x'(W) = 0$, all these inequalities must be tight. Thus we obtain $|x_i| = |x_{i+1}|$ for $i \in [r - 1]$ and $x_\infty = \phi(x)$. For any $P_i \subseteq V_i$ and $P_{i+1} \subseteq V_{i+1}$, we have

$$\begin{aligned} x_i(P_i) - x_{i+1}(V_{i+1} \setminus P_{i+1}) &= \partial x'(P_i^+ \cup (V_{i+1} \setminus P_{i+1})^-) \\ &\leq f(P_i^+ \cup (V_{i+1} \setminus P_{i+1})^-) = \lambda(P_i, P_{i+1}), \end{aligned}$$

which means $(x_i, x_{i+1}) \in L_i$, for $i \in [r - 1]$. Thus $x = (x_1, \dots, x_r)$ is a polylinking flow in G . □

The maximum flow in the polylinking network G can be obtained by finding a submodular flow x in D maximizing $x(a^*)$. This is the maximum submodular flow problem, which can be solved in essentially the same running time as submodular function minimization [8]. Once the maximum flow value ϕ^* is computed, we can set the lower bound $\underline{c}(a^*) = \phi^*$, and seek for a submodular flow minimizing the cost $\sum_{a \in A} d(a)x(a)$. This way, we can efficiently find a minimum-cost maximum flow in a polylinking network. Using an algorithm of Fleischer and Iwata [8] a minimum-cost maximum submodular flow can be found in $O(n^5 \log(nM))$ arithmetic operations and calls to the submodular function f as defined by (2), where $n = |V|$, and M is an upper bound on the function f , assuming f takes integral values.

3.4 Polylinking flow polytope

In this section we introduce the polylinking flow polytope which is the set of all flows of a polylinking network. We show that the polylinking flow polytope is integral if the underlying polylinking systems are integral.

Let $G = (V, L)$ be a polylinking network with r layers. We define the *polylinking flow polytope* $\text{PFP}(G)$ of G as follows.

$$\text{PFP}(G) = \left\{ \begin{array}{l} x_i(P_i) - x_{i+1}(P_{i+1}) \leq \lambda_i(P_i, V_{i+1} \setminus P_{i+1}) \quad \forall i \in [r - 1], \\ \forall P_i \subseteq V_i, \\ \forall P_{i+1} \subseteq V_{i+1} \\ \forall i \in [r - 1] \\ |x_i| = |x_{i+1}| \\ (x_1, \dots, x_r) \in \mathbb{R}_+^{V_1} \times \dots \times \mathbb{R}_+^{V_r} \end{array} \right.$$

The inequalities of the polytope can be interpreted as follows. The flow contained at P_i , i.e., $x_i(P_i)$, has to be sent to the next layer V_{i+1} and thus cannot be larger than the flow at P_{i+1} plus the maximum amount of flow that can be sent from P_i to $V_{i+1} \setminus P_{i+1}$. Hence,

$$x_i(P_i) \leq x_{i+1}(P_{i+1}) + \lambda_i(P_i, V_{i+1} \setminus P_{i+1}).$$

Theorem 5 *Let $G = (V, L)$ be a polylinking network with r layers. The polytope $\text{PFP}(G)$ is the set of all polylinking flows of G .*

Proof By definition of a polylinking flow, a vector $x = (x_1, \dots, x_r) \in \mathbb{R}_+^{V_1} \times \dots \times \mathbb{R}_+^{V_r}$ is a flow in G if $|x_i| = |x_{i+1}|$ for $i \in [r - 1]$ and $(x_i, x_{i+1}) \in L_i$. By Proposition 1 we have that $(x_i, x_{i+1}) \in L_i$ if and only if x_i and x_{i+1} satisfy the inequalities of $\text{PFP}(G)$. □

Similar to the case of classical flows, we have the following integrality property for polylinking networks.

Theorem 6 *If $G = (V, L)$ is an integral polylinking network then $\text{PFP}(G)$ is integral.*

Proof If all the polylinking functions λ_i are integral, the submodular function f defined in (2) is also integral. By Theorems 4 and 5, the polytope $\text{PFP}(G)$ is equal to the submodular flow polytope $\text{SFP}(D, f)$ which is known to be integral if f is integral [15]. □

4 The ADT model as a linking network

In this section we consider a relatively recent information flow model for wireless networks, which we call the *ADT model*, that was introduced in [2] and show that it is a linking network. So far, not many results are known for the ADT model. In [3] a max-flow min-cut result was proven and in [1] a polynomial-time algorithm for

finding a maximum flow was presented. In parallel to this work, Sadegh Tabatabaei Yazdi and Savari [12] developed another polynomial algorithm for finding a maximum flow in an ADT model, which is based on an extended Rado-Hall transversal theorem. The ADT model is of special interest as an example for polylinking flows, since the results derived for general polylinking systems do not only generalize the known results but also lead to new properties and algorithms. Furthermore, we present a compact representation of the ADT model as a polylinking network. Using this compact representation allows us to present a faster algorithm for finding a maximum flow in the ADT model.

4.1 The ADT flow model

The ADT flow model was introduced as a linear deterministic approximation of networked Gaussian channels in a wireless setting. It takes into account broadcasting and interference effects. The random noise of a Gaussian channel is approximated by introducing between every pair of relays a threshold on the number of signals that can be transmitted from one relay to the other. A principal motivation for introducing the ADT flow model was to approximately determine the capacity of networked Gaussian channels. Whereas very little is known of how to compute the capacity of nontrivial combinations of Gaussian channels, in the ADT model the capacity can be determined in polynomial time. In the following we introduce the ADT flow model as presented in [2]. For additional information about the relation between the ADT flow model and wireless networks with Gaussian channels we refer the reader to [2].

Given is a collection of *nodes* $\mathcal{N} = \{N^1, \dots, N^q\}$, also called *relays*, each of which is a set of $2b$ vertices for some global constant $b \in \mathbb{N}$. For each node $N \in \mathcal{N}$, its vertices are partitioned into two groups of b vertices, called *outputs* and *inputs*. For a node $N^i \in \mathcal{N}$, we denote by $\{v_1^i, \dots, v_b^i\}$ its inputs and by $\{w_1^i, \dots, w_b^i\}$ its outputs. We call the set of all vertices V , the set of all outputs O and the set of all inputs I . The nodes are partitioned into layers T_1, \dots, T_r , where $T_1 = \{N^1\}$ and $T_r = \{N^q\}$. The node N^1 is called *sender* and N^q is called *receiver*. The goal is to send signals in some fixed finite field \mathbb{F}_l from the sender to the receiver. Every input can send a signal to outputs of nodes in the next layer. How the signals are sent from the inputs of nodes in one layer to the outputs of the nodes in the next layer is described by a set of arcs $A \subseteq I \times O$ given as follows. For every pair of nodes N^i, N^j such that N^i is in the layer immediately preceding the layer containing N^j , a number $n_{ij} \in \{0, \dots, b\}$ is given and the arcs of A connecting inputs of N^i with outputs of N^j are given by $\left\{ \left(v_1^i, w_{b+1-n_{ij}}^j \right), \dots, \left(v_{n_{ij}}^i, w_b^j \right) \right\}$. If $n_{ij} = 0$, then there are no arcs connecting inputs of N_i with outputs of N_j . The constants n_{ij} bounds the number of signals that relay N^j can receive from relay N^i . Thus, a small value for n_{ij} models that signals sent from relay N^i to N^j suffer a high noise. We call the triple $G = (V, A, \mathcal{N})$ an *ADT network*. See Fig. 1 for an illustration of an ADT network.

We denote by I_i the inputs in layer i and O_i the outputs in layer i , i.e., $I_i = I \cap (\cup_{N \in T_i} N)$, $O_i = O \cap (\cup_{N \in T_i} N)$. See Fig. 2 for an illustration.

Signals are sent in the following way. At each input w_1^1 of the sender, a signal $s \in \mathbb{F}_l$ can be broadcast to the next layer. The signal is sent to all outputs in the next

Fig. 1 Example of an ADT network with $q = 6$. The rectangles represent nodes. The vertices contained in the left half of each node are outputs and the ones in the right half are inputs. In this example we have $n_{12} = 5, n_{13} = 2, n_{24} = 5, n_{25} = 3, n_{34} = 4, n_{35} = 2, n_{46} = 3, n_{56} = 4$

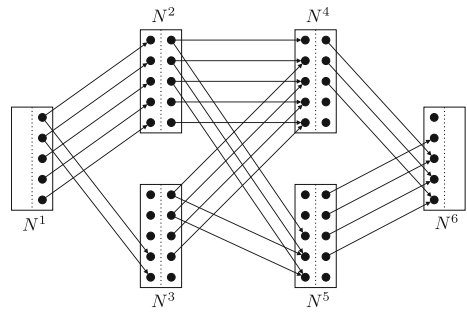
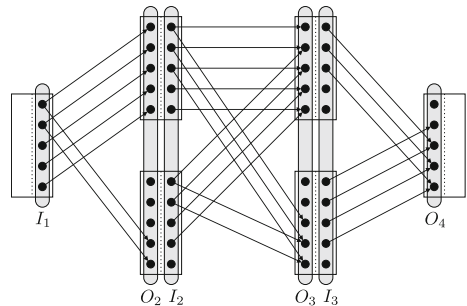


Fig. 2 Representation of the output sets O_i and input sets I_i



layer that are connected by an arc to w_l^1 . This models broadcasting, i.e., a signal in the wireless network cannot be directed towards a single particular output but may reach other outputs as a side effect. If an output receives signals from different inputs, interference between the received signals happens which is modeled as follows. The output receives the sum in \mathbb{F}_l of the incoming signals. Instead of the sum, any other linear combination of the incoming signals with positive coefficients can be chosen without changing the model or the algorithms to be presented in any significant way. Every node receiving signals at its outputs can resend them over its inputs. A signal received at a particular output can be resent over any input of the same node. However, every input can be used at most once. Not every output has to be linked to an input and thus not every input has to be used. The assignment of inputs of a given node to its outputs is called *wiring*. Finally, the receiver receives at its outputs a set of linear combinations of the signals sent by the sender. The signals have to be sent in such a way that the receiver can decode the original signals, i.e., if k signals are sent from the sender, then to properly decode the signals, the receiver needs to receive k signals that are linearly independent combinations (over \mathbb{F}_l) of the k signals sent by the sender. The task is to send the largest number of decodable signals from the sender to the receiver. As observed in [2], decodability of the signal does not depend on the exact wiring inside the nodes but only on the set of outputs and inputs that are used inside the nodes. Since the exact wiring is not important, a flow in an ADT network can be defined as the set of vertices used for receiving or sending signals as follows. For some set $U \subseteq V$, we denote by $M[U] \in \mathbb{F}_l^{(U \cap I) \times (U \cap O)}$ the matrix such that for $i \in U \cap I$ and $j \in U \cap O$, $M[U]_{ij} = 1$ if $(i, j) \in A$ and $M[U]_{ij} = 0$ otherwise. A *flow* F in the ADT network $G = (V, A, N)$ is a subset of the vertices V such that:

- (i) For every node $N \in \mathcal{N}$, $|F \cap N \cap O| = |F \cap N \cap I|$, i.e., the number of used outputs in every node equals the number of used inputs.
- (ii) The matrix $M[F]$ is nonsingular (over \mathbb{F}_l).

The first condition makes sure that for every node $N \in \mathcal{N}$, it is possible to wire the signals received at its outputs $N \cap O \cap F$ to the inputs $N \cap I \cap F$. The second condition guarantees that the signals received at the outputs $F \cap N^q$ of the receiver are linearly independent combinations of the signals sent by the sender and thus are decodable. Since the network is layered, the second condition can also be restated as a condition that has to hold for every pair of consecutive layers.

(ii') For every $i = 1, \dots, r - 1$, $M[F \cap (I_i \cup O_{i+1})]$ has full rank.

The value of a flow F is measured by $|F \cap N^1| = |F \cap N^q|$. For finding a flow of maximum value in the ADT model, Amaudruz and Fragouli give a combinatorial algorithm with running time bounded by $O(n|A|(\phi^*)^5)$ [1, Proposition 3.3], where ϕ^* is the capacity of the network, i.e. the maximum flow value, and $n = |I \cup O|$.

4.2 Reduction to linking networks

Consider an ADT network with q layers of nodes. The ADT model is represented by a linking network defined over the $2q - 2$ layers $I_1, O_2, I_2, \dots, O_q$ using two types of linking systems. For $i \in [q - 1]$, the linking system that links I_i to O_{i+1} is the linking system induced by the matrix $M[I_i \cup O_{i+1}]$. This ensures that a set of linear independent signals that are sent from one layer of inputs to the next layer of outputs results in linear independent signals at the outputs. Thus the condition ii) of an ADT flow is satisfied. For $i \in \{2, \dots, q\}$, the set O_i is linked to I_i by a linking system induced by the bipartite graph that contains an edge between a vertex $v \in O_i$ and $w \in I_i$ if v and w are in the same node. These linking systems enforce that a linking flow satisfies property i) of an ADT flow. Hence a linking flow in this linking network is indeed an ADT flow in the corresponding ADT model and vice versa.

5 An algorithm for linking networks

5.1 Finding a maximum flow via matroid partition

We now show a reduction that allows for solving linking flow problems using any matroid partition algorithm. Given any linking network $G = (V, \Lambda)$ with r layers, consider the matroid M^* defined as the union of the matroids M_{Λ_i} for $i \in [r - 1]$. Recall that M_{Λ_i} is the matroid whose bases are given by the set \mathcal{B}_{Λ_i} , which is defined in Proposition 5, i.e. for every linked pair $(P_i, P_{i+1}) \in \Lambda_i$ there is a base $P_i \cup (V_{i+1} \setminus P_{i+1})$ in \mathcal{B}_{Λ_i} .

This is a matroid whose ground set is the union $\cup_{i=1}^r V_i$ of the ground sets of the M_{Λ_i} 's, and whose independent sets are $\{\cup_{i=1}^{r-1} I_i : I_i \in \mathcal{I}(M_{\Lambda_i})\}$ where $\mathcal{I}(M)$ denotes the family of independent sets of matroid M . The resulting union M^* is indeed a matroid, see [15, Chap. 42] for a proof and discussion. The independent sets of $\mathcal{I}(M^*)$ are called *partitionable*.

The connection between the maximum flow problem in G and independent sets in M^* is highlighted in the following theorem.

Theorem 7 *Let M^* be the matroid described above corresponding to a linking network $G = (V, \Lambda)$. Then*

1. *Given a flow $F = (F_1, \dots, F_r)$ in G , we have*

$$F_1 \cup \left(\bigcup_{i=2}^{r-1} V_i \right) \cup (V_r \setminus F_r) \in \mathcal{I}(M^*). \tag{3}$$

2. *Given any base B of M^* such that*

$$B \supseteq \bigcup_{i=2}^{r-1} V_i, \tag{4}$$

and expressed as $B = \bigcup_{i=1}^{r-1} I_i$ with $I_i \in \mathcal{I}(M_{\Lambda_i})$ one can derive a flow of G by letting:

$$F_i = \begin{cases} I_i \cap V_i & i \in [r - 1] \\ V_r \setminus I_r & i = r. \end{cases} \tag{5}$$

In particular, $F_1 = B \cap V_1$.

Proof (1) Given the flow $F = (F_1, \dots, F_r)$, we know that $F_i \cup (V_{i+1} \setminus F_{i+1}) \in \mathcal{I}(M_{\Lambda_i})$ for $i \in [r - 1]$, and therefore (3) holds.

- (2) Consider a base B of M^* with the required properties. We claim that each I_i is a base of M_{Λ_i} . Since all bases of a matroid have the same cardinality, this can be verified by showing that there is one basis of M^* that is a disjoint union of bases of M_{Λ_i} , for $i \in [r - 1]$. The set $\bigcup_{i=2}^r V_i \in \mathcal{I}(M^*)$ is such a set since V_{i+1} is a basis of M_{Λ_i} for $i \in [r - 1]$.

Defining F by (5), we immediately get that $F = (F_1, \dots, F_r)$ satisfies the definition of a flow. Furthermore, $F_1 = I_1 \cap V_1 = B \cap V_1$. □

In M^* (as in any (non-disjoint) union of matroids), it is not completely straightforward to test independence of a set. This is the purpose of *matroid partition* algorithms which proceed incrementally. Given disjoint sets $I_i \in \mathcal{I}(M_{\Lambda_i})$, $I = \bigcup_{i=1}^{r-1} I_i$ and $v \notin I$, one fundamental step of a matroid partition algorithm decides whether $I \cup \{v\} \in \mathcal{I}(M^*)$ and if so, finds disjoint $I_i' \in \mathcal{I}(M_{\Lambda_i})$ with $I \cup \{v\} = \bigcup_{i=1}^{r-1} I_i'$. This can be used to find a maximum flow in a linking network. Indeed, using Theorem 7, we can start from $I = \bigcup_{i=1}^{r-2} V_{i+1} \in \mathcal{I}(M^*)$ (I consists of all elements except the first and last layers) and repeatedly first check whether each element of V_1 can be added to our current set $I \in \mathcal{I}(M^*)$, and then do the same for V_r . This results in a base of M^* satisfying (4) and having as many elements of V_1 as possible; therefore, as in the proof of the theorem, we can extract a maximum flow. Observe that, for our initial set I , we have a trivial decomposition of it into $\bigcup_{i=1}^{r-1} I_i$ by taking $I_i = V_{i+1}$ for $i \in [r - 2]$ and $I_{r-1} = \emptyset$.

We need now to discuss how the fundamental step in a matroid partition algorithm can be performed. There exist classical algorithms for the fundamental step,

see [15, Section 42.3]. This boils down to constructing a digraph on $\cup_{i=1}^r V_i$ and finding a shortest directed path in it. The arc set of this digraph is $\cup_{i=1}^{r-1} E_i$ with

$$E_i = \{(s, t) \mid s \notin I_i, t \in I_i, I_i \cup \{s\} \setminus \{t\} \in \mathcal{I}(M_{\Lambda_i})\} \subset (V_i \cup V_{i+1}) \times (V_i \cup V_{i+1}).$$

For each $i \in [r - 1]$, consider a vertex subset $S_i = \{s \in (V_i \cup V_{i+1}) \setminus I_i \mid I_i \cup \{s\} \in \mathcal{I}(M_{\Lambda_i})\}$. The fundamental step searches for a directed path from $v \notin I$ to any vertex in $\cup_{i=1}^{r-1} S_i$ with minimum number of arcs. If such a directed path exists, then $I \cup \{v\} \in \mathcal{I}(M^*)$, and exchanging the elements along the directed path provides a partition of $I \cup \{v\}$ into disjoint $I_i' \in \mathcal{I}(M_{\Lambda_i})$. Otherwise, $I \cup \{v\} \notin \mathcal{I}(M^*)$. Since, for every $i \in [r - 2]$, we start from a base $I_i = V_i$ of M_{Λ_i} , we have that $S_i = \emptyset$ for $i \in [r - 2]$; only S_{r-1} is non-empty.

For the ADT model, constructing this digraph takes $O(rn^3)$, and this dominates the time to find appropriate directed path in it. As we are performing at most $2n$ fundamental steps, this gives an overall running time of $O(rn^4)$. Using the analysis technique by Cunningham [6], one can show that the total length of the directed paths in the above matroid partition algorithm is $O(rn \log n)$, which leads to an improved running time bound $O(rn^3 \log n)$. Thus, for ADT networks with large capacity, for example $\phi^* = \Theta(n)$, this algorithm is considerably faster than the algorithm of Amaudruz and Fragouli [1], whose running time is bounded by $O(n|A|(\phi^*)^5)$, where $|A|$ is the number of arcs of the ADT model.

5.2 Finding a minimum source-destination cut

We now show how a minimum source-destination cut can be obtained as a by-product of the above matroid partition algorithm.

At the end of the algorithm we get a solution set $B = \cup_{i=1}^{r-1} I_i$ with $I_i \in \mathcal{I}(M_{\Lambda_i})$ which is a base of M^* and corresponds to a maximum flow by Theorem 7. We now consider the digraph for the fundamental step of the matroid partition algorithm that corresponds to this final solution. We know that this digraph has no directed path from $V_1 \setminus I_1$ to V_r ; indeed such a path from $s \in V_1 \setminus I_1$ to $t \in V_r$ would mean that $B \setminus \{t\} \cup \{s\}$ is also a base of M^* implying that the algorithm should have added s when it considered it. In fact, since the vertices in $V_1 \cap I_1$ have no outgoing arcs (see the definition of E_1), there are no directed paths from V_1 to V_r . Let $W \subset \cup_{i=1}^{r-1} V_i$ be the set of all vertices that are reachable from the vertices V_1 in this digraph; thus $W \cap V_r = \emptyset$ and $V_1 \subseteq W$.

Theorem 8 *The set W is a minimum source-destination cut.*

Proof Again let M^* be the union of the matroids M_{Λ_i} for $i \in [r - 1]$ and let M^* be the restriction of the matroid M^* to the ground set $\cup_{i=1}^{r-1} V_i$, i.e., a set $I \subseteq \cup_{i=1}^{r-1} V_i$ is independent in M^* if it is independent in M^* . Since we considered vertices in V_1 prior to those in V_r , $B \setminus V_r$ is a base of M^* . Observe that W also corresponds to the set of vertices reachable from V_1 in the digraph for the fundamental step of the matroid partition algorithm corresponding to the matroid M^* . It is well-known in matroid theory (see for example [5]) that W corresponds to an optimality certificate, i.e., it satisfies

$$|B| = \rho_-^*(\cup_{i=1}^{r-1} V_i) = |(\cup_{i=1}^{r-1} V_i) \setminus W| + \sum_{i=1}^{r-1} \rho_{\Lambda_i}(W), \tag{6}$$

where ρ_-^* is the rank function of the matroid M_-^* and ρ_{Λ_i} is the rank function of M_{Λ_i} for $i \in [r - 1]$. Let ϕ^* be the maximum flow value of the network. By Theorem 7, we have

$$\rho_-^*(\cup_{i=1}^{r-1} V_i) = |\cup_{i=2}^{r-1} V_i| + \phi^*. \tag{7}$$

Let $W_i = W \cap V_i$ for $i \in [r]$. Notice that $W_1 = V_1$ and $W_r = \emptyset$. By Proposition 6 we have

$$\sum_{i=1}^{r-1} \rho_{\Lambda_i}(W) = \sum_{i=1}^{r-1} (\lambda_i(W_i, V_{i+1} \setminus W_{i+1}) + |W_{i+1}|) = \kappa(W) + \sum_{i=2}^{r-1} |W_i|.$$

Combining this result with (6) and (7), we obtain

$$\phi^* = |V_1 \setminus W_1| + \kappa(W) = \kappa(W),$$

which implies that W is a minimum cut of the linking network. □

6 Compact formulation of the ADT model

6.1 Polylinking systems obtained by aggregation

Let (V_1, V_2, L) be a polylinking system with polylinking function λ , let $D_1 = \{V_1^1, \dots, V_1^p\}$ be a partition of V_1 and $D_2 = \{V_2^1, \dots, V_2^q\}$ be a partition of V_2 . The *polylinking system obtained from (V_1, V_2, L) by aggregating D_1 and D_2* is the polylinking system (D_1, D_2, L') defined by the following polylinking function λ' ,

$$\lambda'(P_1, P_2) = \lambda \left(\bigcup_{Y \in P_1} Y, \bigcup_{Y \in P_2} Y \right) \quad \forall P_1 \subseteq D_1, P_2 \subseteq D_2.$$

It follows easily from the properties of the polylinking function λ that λ' is a polylinking function and thus that (D_1, D_2, L') is a polylinking system. The following proposition shows a basic relation between linked vectors in (V_1, V_2, L) and linked vectors in (D_1, D_2, L') .

Proposition 10 *Let (V_1, V_2, L) be a polylinking system, $D_1 = \{V_1^1, \dots, V_1^p\}$ a partition of V_1 and $D_2 = \{V_2^1, \dots, V_2^q\}$ a partition of V_2 . Let (D_1, D_2, L') be the polylinking system obtained from (V_1, V_2, L) by aggregating D_1 and D_2 .*

- (i) Let $(x_1, x_2) \in L$ and let $(y_1, y_2) \in \mathbb{R}^{D_1} \times \mathbb{R}^{D_2}$ be defined by $y_1(V_1^j) = x_1(V_1^j)$ for all $j \in [p]$ and $y_2(V_2^j) = x_2(V_2^j)$ for all $j \in [q]$. Then $(y_1, y_2) \in L'$.
- (ii) Let $(y_1, y_2) \in L'$. Then there exists $(x_1, x_2) \in L$ such that $y_1(V_1^j) = x_1(V_1^j)$ for all $j \in [p]$ and $y_2(V_2^j) = x_2(V_2^j)$ for all $j \in [q]$. Furthermore, if (V_1, V_2, L) is an integral polylinking system and (y_1, y_2) is integral, then (x_1, x_2) can also be chosen to be integral.

The first statement of the above proposition follows easily from the definition of aggregation. The second statement can be shown by polymatroid intersection.

6.2 Compactification to polylinking networks

The above representation of the ADT network as a linking network can be made more compact using the following representation as a polylinking network. For every node, we aggregate its inputs to one node and its outputs to another node. The linking systems used in the original representation of the ADT network are thus transformed into polylinking systems as described in Sect. 6.1. By Proposition 10, there is a one-to-one correspondence between linking flows in the linking network before aggregation and the aggregated polylinking network. Furthermore, in the aggregated network, the polylinking systems combining output vertices to input vertices are trivial ones that simply send for every node all the flow of the aggregated output vertex to the aggregated input vertex. Thus, they can simply be eliminated by identifying for every node the aggregated output vertex with the aggregated input vertex.

As an implication of the equivalence between the aggregated representation of an ADT network and the original ADT network we get the following result, which was already stated in [3].

Proposition 11 *A minimum source-destination cut in an ADT network can always be chosen such that for every node, all vertices contained in the node are on the same side of the cut.*

6.3 Evaluating the polylinking function

In this section, we provide an efficient algorithm for computing the values of the polylinking functions that arise in the compact formulation of the ADT model.

We start by giving some more details of the polylinking functions encountered after compactification by using as an example the ADT network shown in Figs. 1 and 2 of Sect. 4. More precisely, we want to discuss the polylinking function that links the second and third layer of nodes, i.e., nodes $\{N^2, N^3\}$ and nodes $\{N^4, N^5\}$ in the compact formulation. Below, we present the 0/1-matrix $M[I_2 \cup O_3]$ as defined in Sect. 4, i.e., the rows of $M[I_2 \cup O_3]$ correspond to the vertices in I_2 , the columns correspond to the vertices in O_3 , and there is a one at entry $(i, j) \in I_2 \times O_3$ if and only if there is an arc from i to j in the ADT network. The rows of the matrix can be

naturally partitioned into two parts $N^2 \cap I_2$ and $N^3 \cap I_2$, depending on whether the corresponding inputs belong to N^2 or N^3 . Similarly, the columns can be partitioned into $N^4 \cap O_3$ and $N^5 \cap O_3$.

$$\begin{array}{l}
 N^2 \cap I_2 \\
 N^3 \cap I_2
 \end{array}
 \left\{
 \begin{array}{c}
 \overbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}^{N^4 \cap O_3} \quad \overbrace{\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}^{N^5 \cap O_3} \\
 \hline
 \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}
 \end{array}
 \right.$$

The polylinking function $\lambda : 2^{\{N^2, N^3\}} \times 2^{\{N^4, N^5\}} \rightarrow \mathbb{Z}_+$ that links the aggregated layers $\{N^2, N^3\}$ and $\{N^4, N^5\}$ in the compacted ADT network is then given as follows: for $P \subseteq \{N^2, N^3\}$, $Q \subseteq \{N^4, N^5\}$, $\lambda(P, Q)$ is the rank of the submatrix of $M[I_2 \cup O_3]$ defined on the rows corresponding to P and the columns corresponding to Q . Notice that in the uncompactified version of the ADT network, the task of evaluating the linking function corresponded to determining the rank of a given square submatrix of $M[I_2 \cup O_3]$ which was build by two arbitrary subsets of rows and columns of the same size. In contrast, in the compactified version, only “block-wise” submatrices have to be considered, where the blocks are defined by the nodes. In the following, we discuss a fast method, based on FFT, to evaluate polylinking functions such as λ that are encountered in the compact formulation.

Let F be a field and $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_b)$ be a sequence of b elements in F . We denote by $T(\sigma)$ the $b \times b$ matrix defined by

$$T(\sigma)_{ij} = \begin{cases} \sigma_{b-j+i} & (i \leq j) \\ 0 & (i > j). \end{cases}$$

We call such a matrix $T(\sigma)$ an upper-triangular Toeplitz matrix (or a UTT-matrix).

For a pair of sequences σ and τ of length b , the convolution $\sigma * \tau$ is defined by

$$(\sigma * \tau)_k = \sum_{i=k}^b \sigma_{b-i+k} \tau_i$$

for $k = 1, 2, \dots, b$. Then we have $T(\sigma)T(\tau) = T(\sigma * \tau)$. Thus the product of two UTT-matrices is again a UTT-matrix. If $T(\sigma)$ is nonsingular, the inverse matrix $T(\sigma)^{-1}$ is also a UTT-matrix.

For a sequence σ of length b , the largest number ℓ with $\sigma_\ell \neq 0$ is called the effective length of σ and denoted by $\ell(\sigma)$. Note that $\ell(\sigma) = \text{rank } T(\sigma)$ holds.

Let \mathbb{F}_l be a finite field and let M be a matrix with elements in \mathbb{F}_l that can be partitioned into blocks:

$$M = \begin{pmatrix} M_{11} & \cdots & M_{1v} \\ \vdots & \ddots & \vdots \\ M_{\mu 1} & \cdots & M_{\mu v} \end{pmatrix},$$

where each block $M_{\alpha\beta}$ is a $b \times b$ UTT-matrix. We now discuss how to compute the rank of M over \mathbb{F}_l efficiently. In the ADT model, the polylinking function values are given by the ranks of matrices in this form.

Suppose without loss of generality that M_{11} attains the maximum rank ℓ among the blocks. We can eliminate $M_{\alpha 1}$ for $\alpha = 2, \dots, \mu$ by row operations as follows. For each $\alpha = 1, 2, \dots, \mu$, let N_α be the submatrix of $M_{\alpha 1}$ indexed by the first ℓ rows and the last ℓ columns. Then the entries in the remaining part of $M_{\alpha 1}$ are all zero. For $\alpha = 2, \dots, \mu$, let S_α be the $b \times b$ matrix in the form of

$$S_\alpha := \begin{pmatrix} N_\alpha N_1^{-1} & O \\ O & O \end{pmatrix}.$$

We then construct the matrix

$$M' = \begin{pmatrix} M_{22}' & \cdots & M_{2v}' \\ \vdots & \ddots & \vdots \\ M_{\mu 2}' & \cdots & M_{\mu v}' \end{pmatrix}.$$

by $M_{\alpha\beta}' := M_{\alpha\beta} - S_\alpha M_{1\beta}$. The resulting matrix M' satisfies $\text{rank } M = \ell + \text{rank } M'$. Each block of M' is a UTT-matrix, and we apply the same procedure recursively to M' . Thus we obtain an algorithm for computing the rank of M .

We now discuss the time complexity of this algorithm. For simplicity, we henceforth assume $\mu = v$. The algorithm requires $O(\mu^3)$ multiplications and $O(\mu)$ inversions of UTT-matrices. The matrix M can be encoded by μ^2 sequences of length b .

The multiplication of UTT-matrices reduces to convolution, which can be performed in $O(b^2)$ time by a straightforward implementation. The inversion of a UTT-matrix can be done in $O(b^2)$ time as well.

To devise a faster implementation of the above algorithm, we consider the use of the fast Fourier transformation (FFT). In fact, it is a standard technique to use FFT for computing the convolution over the reals or complex numbers in $O(b \log b)$ time. However, the standard FFT over the reals or complex numbers does not trivially translate to any finite field. In [11] a FFT working on any finite field \mathbb{F}_l , was presented, which has a running time of $O(bm \log(bm))$, where m is the exponent of the characteristic of \mathbb{F}_l , i.e., $l = p^m$ with p prime. In particular, when accepting the typical assumption that the number of possible signals is considered to be constant, the algorithm runs in $O(b \log b)$.

As for the inversion of UTT-matrices, there has been developed a fast divide-and-conquer algorithm using FFT [4], which runs in $O(b \log b)$ time on any field, on which a FFT can be performed in $O(b \log b)$ time.

To sum up, the time complexity of the entire algorithm for computing the rank of M is $O(\mu^3 b \log b)$, when the number of possible signals is considered to be a constant.

7 Conclusion

A layered flow model based on polylinking systems has been introduced that generalizes the classical flow model of Ford and Fulkerson restricted to acyclic networks and the ADT flow model used in the context of wireless information flows. Because of the abstract nature of polylinking systems, the proposed model is very general. Despite its generality, it is possible to find efficiently a maximum flow, a minimum cut, and a minimum cost flow if it is possible to evaluate in polynomial time the polylinking functions that underly the network. The algorithms used for the above optimization problems rely on submodular flow. Furthermore, several properties of polylinking networks have been presented, including the max-flow min-cut theorem and the integrality of the flow polytope.

The reduction to linking networks leads to an efficient algorithm for finding a maximum flow and a minimum cut in the ADT flow model based on matroid partition. This algorithm is faster than the combinatorial algorithm recently presented by Amaudruz and Fragouli [1]. Furthermore, introducing the notion of aggregation of polylinking systems, we present a more compact polylinking network formulation of the ADT model. With the aid of an effective use of FFT in evaluating the polylinking functions, this formulation leads to another efficient method of analyzing the ADT model based on submodular flow algorithms.

Acknowledgments We thank the anonymous referees for many helpful comments and suggestions.

References

1. Amaudruz, A., Fragouli, C.: Combinatorial algorithms for wireless information flow. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 555–564 (2009)
2. Avestimehr, A.S., Diggavi, S.N., Tse, D.N.C.: A deterministic approach to wireless relay networks. In: Proceedings of the Allerton Conference on Communication, Control, and Computing (2007). http://licos.epfl.ch/index.php?p=research_projWNC
3. Avestimehr, A.S., Diggavi, S.N., Tse, D.N.C.: Wireless network information flow. In: Proceedings of the Allerton Conference on Communication, Control, and Computing (2007). http://licos.epfl.ch/index.php?p=research_projWNC
4. Commenges, D., Monsion, M.: Fast inversion of triangular Toeplitz matrices. *IEEE Trans. Autom. Control* **AC-29**, 250–251 (1984)
5. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: *Combinatorial Optimization*. Wiley, New York (1998)
6. Cunningham, W.H.: Improved bounds for matroid partition and intersection. *SIAM J. Comput.* **15**, 948–957 (1986)
7. Edmonds, J., Giles, R.: A min-max relation for submodular functions on graphs. *Ann. Discrete Math.* **1**, 185–204 (1977)

8. Fleischer, L., Iwata, S.: Improved algorithms for submodular function minimization and submodular flow. In: Proceedings of the 32nd ACM Symposium on Theory of Computing, pp. 107–116 (2000)
9. Goemans, M.X., Iwata, S., Zenklusen, R.: An algorithmic framework for wireless information flow. In: Forty-Seventh Annual Allerton Conference on Communication, Control, and Computing (2009)
10. Kung, J.P.S.: Bimatroids and invariants. *Adv. Math.* **30**, 238–249 (1978)
11. Preparata, F.P., Sarwate, D.V.: Computational complexity of Fourier transforms over finite fields. *Math. Comput.* **31**(139), 740–751 (1977)
12. Yazdi, S.M.S.T., Savari, S.A.: A max-flow/min-cut algorithm for a class of wireless networks. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10, pp. 1209–1226. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2010)
13. Schrijver, A.: Matroids and linking systems. Ph.D. thesis, Mathematisch Centrum (1978)
14. Schrijver, A.: Matroids and linking systems. *J. Comb. Theory Ser. B* **26**(3), 349–369 (1979)
15. Schrijver, A.: *Combinatorial Optimization—Polyhedra and Efficiency*. Springer, Berlin (2003)