On the Complexity of Compressing Two Dimensional Routing Tables with Order

Frédéric Giroire, Frédéric Havet, Joanna Moulierac

▶ To cite this version:

HAL Id: hal-01686641

https://hal.inria.fr/hal-01686641

Submitted on 17 Jan 2018

On the Complexity of Compressing Two Dimensional Routing Tables with Order

**Frédéric Giroire · Frédéric Havet ·
Joanna Moulierac**

Abstract Motivated by routing in telecommunication network using Software Defined Network (SDN) technologies, we consider the following problem of finding short routing lists using aggregation rules. We are given a set of communications $\mathcal{X}$, which are distinct pairs $(s,t) \subseteq S \times T$, (typically S is the set of sources and T the set of destinations), and a port function $\pi : \mathcal{X} \to P$ where P is the set of ports. A *routing list* $\mathcal{R}$ is an ordered list of triples which are of the form (s,t,p), $(*,t,p)$, $(s,*,p)$ or $(*,*,p)$ with $s \in S$, $t \in T$ and $p \in P$. It *routes* the communication (s,t) to the port $r(s,t) = p$ which appears on the first triple in the list $\mathcal{R}$ that is of the form (s,t,p), $(*,t,p)$, $(s,*,p)$ or $(*,*,p)$. If $r(s,t) = \pi(s,t)$, then we say that (s,t) is *properly routed* by $\mathcal{R}$ and if all communications of $\mathcal{X}$ are properly routed, we say that $\mathcal{R}$ *emulates* $(\mathcal{X},\pi)$. The aim is to find a shortest routing list emulating $(\mathcal{X},\pi)$. In this paper, we carry out a study of the complexity of the two dual decision problems associated to it. Given a set of communication $\mathcal{X}$, a port function π and an integer k, the

Frédéric Giroire
Université Côte d'Azur, CNRS, Inria, I3S
Address: Inria Sophia Antipolis, Project-team COATI, 2004 route des Lucioles, B.P. 93, F-06902 Sophia Antipolis Cedex, France
Tel.: +33-4-92 38 50 98, Fax: +33-4-4 89 73 24 00
E-mail: frederic.giroire@cnrs.fr

Frédéric Havet
Université Côte d'Azur, CNRS, Inria, I3S
E-mail: frederic.havet@cnrs.fr

Joanna Moulierac
Université Côte d'Azur, CNRS, Inria, I3S
E-mail: joanna.moulierac@unice.fr

first one called Routing List (resp. the second one, called List Reduction) consists in deciding whether there is a routing list emulating $(\mathcal{X}, \pi)$ of size at most k (resp. $|\mathcal{X}| - k$). We prove that both problems are NP-complete. We then give a 3-approximation for List Reduction, which can be generalized to higher dimensions. We also give a 4-approximation for Routing List in the fundamental case when there are only two ports (i.e. $|P| = 2$), $\mathcal{X} = S \times T$ and $|S| = |T|$.

Keywords Routing · Routing Tables · Order · Priority · Software Defined Networks · Complexity · Approximation Algorithm · Compact Tables.

1 Introduction

Motivation. Software Defined Technology (SDN), e.g. OpenFlow [18] is a new promising approach to operate telecommunication networks. Its promise is to allow to take dynamic routing decisions by decoupling the control plane (the system making decisions) from the data plane (which forwards the packets). This way, a centralized controller receives the data monitored in the system (e.g. load, delay, ...) and then, based on this information, computes appropriate routing decisions. Each time a new flow arrives, the router contacts the controller and waits for the decisions to be pushed into its forwarding table. The routing tables thus are populated with flow-based rules with header informations (source IP, destination IP, ...) → exit port.

In this context, we consider *two-dimensional routing tables* in which the routing decision is not done exclusively on the destination IP addresses, but on the source and destination IP addresses. Indeed, the commonly implemented destination-based routing has its limitations, especially in delivering quality of service which is a goal of SDN paradigm. One suggested remedy is to base the routing decision on additional fields in the packet header. One of the most important field is the source host. For instance, this would permit selective routing to provide a high bandwidth connection between two different sites of a company. Such refined forwarding is part of the next generation Internet design, and falls within the broader scope of layer four packet classification, where packets are routed using arbitrary fields of the packet header [10], [4], [17], [2]. Routers capable of packet classification can implement many advanced services, such as firewall access control, Virtual Private Networks, and quality of service routing, which are all promises of the SDN paradigm.

However, SDN hardware uses specific memory, e.g. TCAM memory [12, 13], which is very expensive and of small size. Thus, the number of entries of the routing tables is limited to only a few thousands [19,23] and grows linearly with the number of flows passing through a router, causing a problem of scalability. It is thus an important area of research to obtain routing using only a limited number of rules per router. [3] studies the problem of choosing routing with a limited number of entries per router using linear programming. Another way to compact the forwarding tables is to use aggregated rules.

This is the approach we consider. With such an aggregation, we can set routing entries such as "(*,destination)→ port" or "(source,*)→ port" or also a default entry such as "(*,*)→ port". For example, [8] studies how to use default ports to reduce the size of routing tables.

In this paper, we consider the *problem of finding the shortest routing list using aggregation rules, that emulates a given routing table.* We study several variants of the problem formally defined in Section 2: ROUTING LIST (given an integer, k, find a routing list of size at most k), LIST REDUCTION (reduce the size of the routing list by at least k rules), with a fixed number of ports k (k-PORT problems), using or not the global triple (WITHOUT-GLOBAL problems).

Contributions. After some preliminaries, we first show in Section 3 that the 2-PORT problems are strongly related to the celebrated FEEDBACK ARC SET problem in an associated bipartite digraph.

In Section 4, we study the complexity of the above problems. We first show that the 1-PORT problems are polynomial-time solvable. In contrast, we show that the 2-PORT problems are NP-complete via a reduction from FEEDBACK ARC SET. As a trivial consequence, ROUTING LIST and LIST REDUCTION, as well as their variants with k-ports for all $k \geq 2$, are NP-complete. This answers an open question of [24]. Similarly to us, the authors considered the problem of determining a compact routing table using aggregation triples that has the same behaviour as the original routing table. The difference with our problem is that their goal was to find what they called a *conflict-free* routing table in which the triples can be taken in any order. On the contrary, as noted above, the order is crucial in our problems. We quote: "The filter compression problem with inconsistent filters [triples], where one uses priority to define best matching filter [triple], is open, and we conjecture that it is NP-complete."

In fact, we prove a stronger result by showing that 2-PORT ROUTING LIST and 2-PORT WITHOUT-GLOBAL ROUTING LIST are NP-complete even when restricted to a full set of communications. A communication set is *full* if it is of the form $S \times T$, and a set of communication triples is *full* if it is a set of communication triples on a full set of communications.

In Section 5, we provide some approximation algorithms. We first describe some heuristics and show that one of them, called Direction-based Heuristic, is a 2-approximation for WITHOUT-GLOBAL LIST REDUCTION and another one, called Global Heuristic, is a 3-approximation for LIST REDUCTION. These algorithms can be generalized for any number of fields $f \geq 2$, leading respectively to an f- and a $(2^f - 1)$-approximations. Next, we focus on the problem with two ports. We provide an approximation algorithm for ROUTING LIST when the set of communication triples are almost full. This algorithm uses a 4-approximate polynomial-time algorithm for FEEDBACK ARC SET in complete bipartite tournaments, due to Van Zuylen [26].

Related Work. Other types of compression of routing tables were considered, e.g. lossy compression [22]. The idea is achieve high compression ratio at the

cost of losing some information. Note that, in this paper, we consider aggregation rules which are a wildcard on a whole field. TCAM memory allows using more expressive wildcards at the granularity of a single bit, such as the classic prefix and range wildcards. There is an extensive litterature on packet classification, that is comparing a packet header to a set of rules, see for example the survey [25]. TCAM memory can efficiently represent prefix rules, but range rules only with difficulty. Recent works propose methods to handle efficiently range rules by identifying classifier properties [21,16]. Note that our hardness results (NP-completeness) extend to more expressive aggregation rules.

2 Problem Modelling and Preliminaries

2.1 Problem Modelling

Modelling. A *communication* in a network is a pair of nodes (s, t). The node s is called the *source* and t the *destination*. We use the source and destination fields in our examples, although our ideas apply to any two prefix fields in Internet protocol networks. A *communication set* is a set of distinct communications, i.e. two communications might have the same source or the same destination, but they cannot have both same source and same destination. A *routing* of a communication (s, t) is a path in the network from s to t. A *routing* of a communication set is the union of routings of its communications.

At each node, there is a set $\mathcal{X}$ of communications whose routing path goes through this node. The node needs to be able to find for each communication (s, t) in $\mathcal{X}$, the port $\pi(s, t)$ that the routing path of (s, t) uses to leave it. An easy way of doing it is to store the set $\mathcal{C} = \mathcal{C}(\mathcal{X}, \pi)$ of all triples $(s, t, \pi(s, t))$ for $(s, t) \in \mathcal{X}$. Such triples are called *communication triples*.

However, such a list might be very large. So we want to reduce it as much as possible (especially when considering the TCAM Memory limitation within SDN) using the $*$ symbol. Hence, in addition, we can use some additional triples, called $*$-*triples*. There are three kinds of $*$-triples:

- t-*destination triple* $(*, t, p)$, meaning that every communication with destination t leaves on port p.
- s-*source triple* $(s, *, p)$, meaning that every communication with source s leaves on port p.
- *global triple* $(*, *, p)$, meaning that all communications leave on port p.

A *routing list* is an ordered list $T_1, \ldots, T_r$ of triples (either communication triples or $*$-triples). It *routes* the communication (s, t) to the port $r(s, t) = p$ which appears on the first triple in the list $\mathcal{R}$ that is of the form (s, t, p), $(*, t, p)$, $(s, *, p)$ or $(*, *, p)$. It is crucial to remark that *using $*$-triples makes the order in the routing list matter*. For example, in the two compacted tables of Table 1, if $(*, 4, \text{Port-4})$ appears in first position, then $(1, 4)$ will not be properly routed through Port-6 as in the initial routing list but through Port-4. Note that the global triple $(*, *, p)$ should appear in the last position, otherwise all the communications are routed to p.

(s, t)	$\pi(s, t)$
$(0, 4)$	Port-4
$(0, 5)$	Port-5
$(0, 6)$	Port-5
$(1, 4)$	Port-6
$(1, 5)$	Port-4
$(1, 6)$	Port-6
$(2, 4)$	Port-4
$(2, 5)$	Port-5
$(2, 6)$	Port-6

(s, t)	$\pi(s, t)$
$(1, 4)$	Port-6
$(1, 5)$	Port-4
$(2, 5)$	Port-5
$(*, 4)$	Port-4
$(0, *)$	Port-5
$(*, 6)$	Port-6

(s, t)	$\pi(s, t)$
$(1, 5)$	Port-4
$(2, 6)$	Port-6
$(1, *)$	Port-6
$(*, 4)$	Port-4
$(*, *)$	Port-5

Table 1 An example with a routing list (on the left) and two minimal routing lists that emulate it, one (in the middle) without the global triple, and the other (on the right) with it.

If $r(s, t) = \pi(s, t)$, then we say that (s, t) is *properly routed* by $\mathcal{R}$. If all communications of $\mathcal{X}$ are properly routed, we say that $\mathcal{R}$ *emulates* $(\mathcal{X}, \pi)$.

$(\mathcal{X}, \pi) \mapsto \mathcal{C}(\mathcal{X}, \pi)$ is an obvious one-to-one mapping between all pairs made of a set of communications and a port function and all sets of communication triples. Therefore, for sake of convenience, we always consider set of communication triples. In particular, we say that $\mathcal{R}$ *emulates* $\mathcal{C}$ if it emulates the corresponding $(\mathcal{X}, \pi)$, that is if each communication of $\mathcal{C}$ is assigned the same port by $\mathcal{C}$ and $\mathcal{R}$. Observe that $\mathcal{R}$ may route more communications than $\mathcal{C}$. For example, if the port of all triples of $\mathcal{C}$ have source s and port p, then the singleton list made of the global triple $(s, *, p)$ emulates $\mathcal{C}$, even if there is not a triple in $\mathcal{C}$ for all communications.

Problems. The problem is then to find the shortest routing list that emulates a given set of communication triples $\mathcal{C}$. We consider first the problems where we are not allowed to use the global triple. Table 1 presents an example of a routing list, together with a minimal routing list with and without the global triple emulating it.

We denote by $\text{rmin}(\mathcal{C})$ the minimum number of triples in a routing list emulating $\mathcal{C}$ without the global triple.

WITHOUT-GLOBAL ROUTING LIST:
Input: A set $\mathcal{C}$ of communication triples and an integer r.
Question: $\text{rmin}(\mathcal{C}) \leq r$?

The *number of saved triples* is $\text{sav}(\mathcal{C}) = |\mathcal{C}| - \text{rmin}(\mathcal{C})$. The complementary problem to WITHOUT-GLOBAL ROUTING LIST is the following.

WITHOUT-GLOBAL LIST REDUCTION
Input: A set $\mathcal{C}$ of communication triples and an integer z.
Question: $\text{sav}(\mathcal{C}) \geq z$?

Let us denote by $\text{rmin}^*(\mathcal{C})$ the minimum number of triples in a routing list with global triples emulating $\mathcal{C}$, and let $\text{sav}^*(\mathcal{C}) = |\mathcal{C}| - \text{rmin}^*(\mathcal{C})$. We study the following variations of WITHOUT-GLOBAL ROUTING LIST and WITHOUT-GLOBAL LIST REDUCTION.

ROUTING LIST:
Input: A set $\mathcal{C}$ of communication triples and an integer r.
Question: $\mathrm{rmin}^*(\mathcal{C}) \leq r$?

LIST REDUCTION
Input: A set $\mathcal{C}$ of communication triples and an integer z.
Question: $\mathrm{sav}^*(\mathcal{C}) \geq z$?

Note that rmin and rmin^* (resp. sav and sav^*) are closely related parameters, in the sense of the two following lemmas.

Lemma 1 *Let $\mathcal{C}$ be a set of communication triples in which n sources appear.*

$$\mathrm{rmin}^*(\mathcal{C}) \leq \mathrm{rmin}(\mathcal{C}) \leq \mathrm{rmin}^*(\mathcal{C}) + n - 1.$$

Proof Clearly, $\mathrm{rmin}^*(\mathcal{C}) \leq \mathrm{rmin}(\mathcal{C})$.

Let $\mathcal{R}$ be a shortest routing list with global triples. Trivially, $\mathcal{R}$ contains at most one global triple, because all communication triples are routed by a global triple. Let $s_1, \ldots, s_n$ be the sources appearing in $\mathcal{C}$. If $\mathcal{R}$ has a global triple $(*, *, p)$, then it can be replaced by the n source triples $(s_i, *, p)$, $1 \leq i \leq n$, to obtain a routing list emulating $\mathcal{C}$ with no global triples. Hence, $\mathrm{rmin}(\mathcal{C}) \leq \mathrm{rmin}^*(\mathcal{C}) + n - 1$. $\square$

Let $M(\mathcal{C})$ be the maximum number of triples of $\mathcal{C}$ with the same port.

Lemma 2 *Let $\mathcal{C}$ be a set of communication triples.*

$$\mathrm{sav}(\mathcal{C}) \leq \mathrm{sav}^*(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C}) + M(\mathcal{C}) - 1.$$

Proof Clearly, $\mathrm{sav}(\mathcal{C}) \leq \mathrm{sav}^*(\mathcal{C})$.

Let us now prove that $\mathrm{sav}^*(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C}) + M(\mathcal{C}) - 1$. Let $\mathcal{R}$ be a shortest routing list with global triples. We have $|\mathcal{R}| = |\mathcal{C}| - \mathrm{sav}^*(\mathcal{C})$. Trivially, $\mathcal{R}$ contains at most one global triple.

If $\mathcal{R}$ contains no global triple, then $\mathrm{sav}(\mathcal{C}) \geq |\mathcal{C}| - |\mathcal{R}|$, and so $\mathrm{sav}^*(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C})$.

Assume now that $\mathcal{R}$ contains a global triple, say τ. Let $\mathcal{R}_\tau$ be the set of triples of $\mathcal{C}$ that are routed by τ. Let $\mathcal{R}'$ be the list obtained from $\mathcal{R}$ by replacing τ by $\mathcal{R}_\tau$ (in any order). Clearly, $\mathcal{R}'$ emulates $\mathcal{C}$ and has no global triple. Hence

$$\mathrm{sav}(\mathcal{C}) \geq |\mathcal{C}| - |\mathcal{R}'| = |\mathcal{C}| - (|\mathcal{R}| + |\mathcal{R}_\tau| - 1) = \mathrm{sav}^*(\mathcal{C}) - (|\mathcal{R}_\tau| - 1).$$

But, by definition, $|\mathcal{R}_\tau| \leq M(\mathcal{C})$. Thus $\mathrm{sav}^*(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C}) + M(\mathcal{C}) - 1$. $\square$

In practice, the number of ports at a node is fixed. Hence it is natural to ask about the complexity of ROUTING LIST and LIST REDUCTION when the number of ports is bounded by a constant. We call k-PORT ROUTING LIST, (resp. k-PORT WITHOUT-GLOBAL ROUTING LIST, k-PORT LIST REDUCTION, k-PORT WITHOUT-GLOBAL LIST REDUCTION), the restriction of ROUTING LIST, (resp. WITHOUT-GLOBAL ROUTING LIST, LIST REDUCTION, WITHOUT-GLOBAL LIST REDUCTION) to the sets of communication triples with at most k ports.

2.2 Standard and canonical routing lists

Let $\mathcal{R} = T_1, \ldots, T_r$ be a routing list, possibly with a global triple. It is *standard* if there exists i such that T_j is a $*$-triple if and only if $j > i$. The following lemma is easy and left to the reader.

Lemma 3 *Let $\mathcal{C}$ be a set of communication triples and $\mathcal{R}$ be a routing list emulating $\mathcal{C}$. Then the routing list $\mathcal{R}'$ obtained from $\mathcal{R}$ by*

- *deleting the useless communication triples (the ones that route no triples),*
- *putting all the communication triples of $\mathcal{R}$ at the beginning and all the $*$-triples of $\mathcal{R}$ at the end, keeping the same order as in $\mathcal{R}$ for the $*$-triples,*

also emulates $\mathcal{C}$ and is standard.

A routing list is *canonical* if it is the concatenation of sublists $\mathcal{B}_1, \ldots, \mathcal{B}_q$, called *blocks*, having the following properties for every $1 \le \ell \le q$:

(i) in $\mathcal{B}_\ell$, there is a unique $*$-triple and it is the last one;
(ii) if the $*$-triple of $\mathcal{B}_\ell$ is an s-source triple (resp. t-destination triple), then all triples of $\mathcal{B}_\ell$ have source s (resp. destination t);
(iii) if $\ell \ne q$, then $\mathcal{B}_\ell$ has no global triple.

Lemma 4 *Let $\mathcal{C}$ be a set of communication triples and $\mathcal{R}$ be a routing list emulating $\mathcal{C}$. One can obtain from $\mathcal{R}$ a canonical routing list $\mathcal{R}'$ emulating $\mathcal{C}$ not longer than $\mathcal{R}$ by successive applications of the following operations:*

- *deleting some triple,*
- *replacing some communication triple (s, t, p) by the triple $(*, t, p)$,*
- *reordering the triples.*

Proof We shall use three operations described in the statement according to the following rules on a routing list $\mathcal{L} = T_1, \ldots, T_r$.

(R1) If there are triples after a global triple, then they are useless, so we delete them.
(R2) If the last triple is a communication triple (s, t, p), then we replace it by $(*, t, p)$.
(R3) Let $i_1, i_2, \ldots, i_q$ be the indices in increasing order of the $*$-triples of $\mathcal{L}$. If for some ℓ, T_{i_ℓ} is an s-source triple (resp. t-destination triple) and there is a communication triple T_i with $i_{\ell-1} < i < i_\ell$ with source distinct from s (resp. destination distinct from t), then we move T_i after T_{i_ℓ}.

It is simple matter to check that if any of the above rules applies to a routing list emulating $\mathcal{C}$, then we obtain a new routing list emulating $\mathcal{C}$ which is not longer that the original one.

Therefore starting from $\mathcal{R}$, as long as one of the rules (R1), (R2) or (R3) applies, we do it. This process must end, because each rule can be applied only a finite number of times. Indeed (R1) decreases the size of the list, (R2) increases the number of $*$-triples, and (R3) decreases the vector $(i_1, i_2, \ldots, i_q)$ of the indices of $*$-triples in the lexicographic order.

Let $\mathcal{R}' = T_1', \ldots, T_r'$ be the routing list obtained at the end of the process. Note that at each step of the process, we have a routing list emulating $\mathcal{C}$ not longer than $\mathcal{R}$ by the above remark. In particular $\mathcal{R}'$ emulates $\mathcal{C}$ and $|\mathcal{R}'| \leq |\mathcal{R}|$. Moreover, none of (R1), (R2), and (R3) applies. Let $i_1, i_2, \ldots, i_q$ be the indices in increasing order of the $*$-triples of $\mathcal{R}'$, and for $1 \leq \ell \leq q$, let $\mathcal{B}_\ell$ be the sublist $T_{i_{\ell-1}+1}, \ldots, T_{i_\ell}$ (with $i_0 = 0$). By definition, the unique $*$-triple in $\mathcal{B}_\ell$ is its last triple. The concatenation $\mathcal{B}_1, \ldots, \mathcal{B}_q$ is $\mathcal{R}'$ for otherwise (R2) would apply. If $\ell \neq q$, then $\mathcal{B}_\ell$ has no global triple, for otherwise (R1) would apply. Finally, if the $*$-triple of $\mathcal{B}_\ell$ is an s-source triple (resp. t-destination triple), then all triples of $\mathcal{B}_\ell$ have source s (resp. destination t), for otherwise (R3) would apply. Hence, $\mathcal{R}'$ is canonical. □

2.3 Feedback arc set

A *feedback arc set* in a digraph D is a set of arcs F such that $D \setminus F$ is acyclic. Given an ordering $\sigma = v_1, \ldots, v_n$ of the vertices of D, an arc $v_i v_j$ of D is *feedback according to σ*, or σ-*feedback*, if $i > j$. A *standard feedback arc set* in D is the set of all σ-feedback arcs for some ordering σ. It is well known that every feedback arc set contains a standard feedback arc set. The minimum size of a feedback arc set in D is denoted $\mathrm{fas}(D)$.

FEEDBACK ARC SET is the following problem:
Input: A digraph D, and an integer k.
Question: Does D have a feedback arc set of cardinality at most k?

This problem is well-known to be NP-complete: it is one of Karp's 21 NP-complete problems [15]. It is also known to be NP-complete for many classes of digraphs including bipartite digraphs. Kann [14] proved that it is APX-complete

A *bipartite tournament* is the orientation of a complete bipartite graph. A bipartite digraph is *balanced* if the two parts of its bipartition have equal size. Guo et al. [9] proved that FEEDBACK ARC SET is NP-complete for bipartite tournaments. It can be easily extended to balanced bipartite tournament.

Lemma 5 FEEDBACK ARC SET *on balanced bipartite tournaments is NP-complete.*

Proof Reduction from FEEDBACK ARC SET on bipartite tournaments. Let D be a bipartite tournament with bipartition (A, B). Without loss of generality, we may assume that $|A| \leq |B|$. Let D' be the balanced bipartite tournament obtained from D by adding $|B| - |A|$ vertices dominating B. The added vertices are sources in D' and so are in no directed cycles. Consequently, $\mathrm{fas}(D) = \mathrm{fas}(D')$. □

3 Relation between 2-PORT ROUTING LIST and FEEDBACK ARC SET PROBLEM

Let $\mathcal{C}$ be a set of communication triples, with source set $S = \{s_1, \ldots, s_n\}$, destination set $T = \{t_1, \ldots, t_m\}$ and port set $\{p_1, p_2\}$. We associate to $\mathcal{C}$ the $n \times m$ matrix $A = A_{\mathcal{C}}$ defined by $a_{i,j} = 1$ if $(s_i, t_j, p_1) \in \mathcal{C}$, $a_{i,j} = -1$ if $(s_i, t_j, p_2) \in \mathcal{C}$, and $a_{i,j} = 0$ otherwise. We also associate to $\mathcal{C}$ the bipartite digraph $D_{\mathcal{C}}$ with vertex set $S \cup T$ in which for all $s \in S$ and all $t \in T$, st is an arc if and only if $(s, t, p_1) \in \mathcal{C}$, and ts is an arc if and only if $(s, t, p_2) \in \mathcal{C}$. Hence, $A_{\mathcal{C}}$ is the biadjacency matrix of the bipartite digraph $D_{\mathcal{C}}$.

Observe that $\mathcal{C} \to A_{\mathcal{C}}$ is a one-to-one correspondence between the sets of communication triples, with source set $S = \{s_1, \ldots, s_n\}$, destination sets $T = \{t_1, \ldots, t_m\}$ and port set $\{p_1, p_2\}$, and the $\{-1, 0, 1\}$-entry $n \times m$ matrices. Similarly, $\mathcal{C} \to D_{\mathcal{C}}$ is a one-to-one correspondence between the sets of communication triples, with source set $S = \{s_1, \ldots, s_n\}$, destination sets $T = \{t_1, \ldots, t_m\}$ and port set $\{p_1, p_2\}$ and the labelled bipartite oriented graphs (an oriented graph is a digraph with no 2-cycles) with vertex set $S \cup T$.

Let us make some easy observations on $\mathcal{D}_{\mathcal{C}}$. In a digraph D, for any vertex v, we denote by $A_D^+(v)$, or simply $A^+(v)$ when D is clear from the context, the set of arcs leaving v. Similalry, we denote by by $A_D^-(v)$, or simply $A^-(v)$, the set of arcs entering v.

Fact 1   *1. The communication triples of $\mathcal{C}$ are in one-to-one correspondence to the arcs of $\mathcal{D}_{\mathcal{C}}$.*
2. If $s \in S$, then $A^+(s)$ corresponds to the set of communication triples with source s and port p_1 and $A^-(s)$ corresponds to the set of communication triples with source s and port p_2.
3. If $t \in T$, then $A^-(t)$ corresponds to the set of communication triples with destination t and port p_1 and $A^+(s)$ corresponds to the set of communication triples with destination t and port p_2.

In view of Fact 1.1, for sake of clarity, we often identify the arcs of $\mathcal{D}_{\mathcal{C}}$ with their corresponding communication triples.

Lemma 6 *Let $\mathcal{C}$ be a set of communication triples with two ports and let $\mathcal{R}$ be a routing list emulating $\mathcal{C}$, possibly with a global triple. The set of communication triples in $\mathcal{R}$ corresponds to a feedback arc set in $D_{\mathcal{C}}$.*

Proof Since in the operations described in Lemma 4 to obtain a canonical routing list, no communication triple is added, it suffices to prove the result for canonical routing lists. Moreover, if $\mathcal{R}$ contains a global triple $(*, *, p)$, then we can replace it by the $(s, *, p)$ for all sources s of $\mathcal{R}$ without increasing the number of communication triples. Consequently, it suffices to prove the result for canonical routing lists without global triples.

We prove the result by induction on the sum of the numbers of sources and destinations of $\mathcal{C}$, the result being vacuously true if $\mathcal{C}$ is empty.

Let $\mathcal{C}$ be a non-empty set of communication triples and set $D = D_{\mathcal{C}}$. Let $\mathcal{R}$ be a canonical routing list with no global triple emulating $\mathcal{C}$ and let $\mathcal{R}'$ be

the set of communication triples of $\mathcal{R}$. For each block $\mathcal{B}_i$ of $\mathcal{R}$, let T_i be the last triple of $\mathcal{B}_i$. Set $\mathcal{B}_i' = \mathcal{B}_1 \cap \mathcal{R}'$. We have $\mathcal{B}_i' = \mathcal{B}_i \setminus \{T_i\}$.

Suppose that T_1 is a source triple, say $(s, *, p)$. Let $p' = \{p_1, p_2\} \setminus \{p\}$. Then all the communication triples with source s and port p' must be routed before T_1, and thus belong to $\mathcal{B}_1$. Hence, if $p' = p_1$ then $A_D^+(s) \subseteq \mathcal{B}_1'$, and if $p' = p_2$ then $A_D^-(s) \subseteq \mathcal{B}_1'$. In particular, there is no cycle through s in $D \setminus \mathcal{B}_1'$. Let $\mathcal{C}'$ be the set of triples obtained from $\mathcal{C}$ by removing the triples with source s. Then $D' = D_{\mathcal{C}'}$ is the digraph $D - s$ and $\mathcal{R} \setminus \mathcal{B}_1$ emulates $\mathcal{C}'$. By the induction hypothesis, $\mathcal{R}' \setminus \{\mathcal{B}_1'\}$, which is the set of communication triples of $\mathcal{R} \setminus \mathcal{B}_1$ corresponds to a feedback arc set in $D - s$. Therefore, $\mathcal{R}'$ is a feedback arc set in D.

Similarly, we get the result if T_1 is a destination triple. $\qquad\square$

The following lemma is a kind of reciprocal to Lemma 6.

Lemma 7 *Let $\mathcal{C}$ be a set of communication triples with n sources and m destinations. If $D_{\mathcal{C}}$ is acyclic, then there is a routing list emulating $\mathcal{C}$ containing at most $n + m - 1$ source or destination triples and no other triples.*

Proof By induction on $n + m$, the result holding trivially when $\mathcal{C}$ has one source and one destination.

Suppose that $\mathcal{C}$ is a set of communication triples with n sources and m destinations, where $n + m \geq 3$. If $D_{\mathcal{C}}$ is acyclic, then it contains a vertex v with in-degree 0. If this vertex is a source (resp. destination), then all the communication triples with source (resp. destination) v have port p_1 (resp. p_2). Hence using first the triple $(v, *, p_1)$ (resp. $(*, v, p_2)$), all communication triples of $\mathcal{C}$ with source (resp. destination) v are routed. Therefore, the set $\mathcal{C}'$ of communication triples that remain to be routed has one source or destination less than $\mathcal{C}$. Moreover $D_{\mathcal{C}'}$ is $D_{\mathcal{C}} - v$, so it is acyclic. Thus, by the induction hypothesis, there is a routing list emulating $\mathcal{C}'$ containing $n + m - 2$ source or destination triples and no other triples. The concatenation of $(v, *, p_1)$ (resp. $(*, v, p_2)$) with this list gives the desired routing list emulating $\mathcal{C}$. $\qquad\square$

The previous two lemmas implies that $\mathrm{rmin}^*(\mathcal{C})$ and $\mathrm{rmin}(\mathcal{C})$ are closely related to $\mathrm{fas}(D_{\mathcal{C}})$.

Corollary 1 *If $\mathcal{C}$ is a set of communication triples with n sources and m destinations, then $\mathrm{fas}(D_{\mathcal{C}}) + 1 \leq \mathrm{rmin}^*(\mathcal{C}) \leq \mathrm{rmin}(\mathcal{C}) \leq \mathrm{fas}(D_{\mathcal{C}}) + n + m - 1$.*

Proof Let $\mathcal{R}$ be a canonical routing list emulating $\mathcal{C}$. By Lemma 6, there are at least $\mathrm{fas}(D_{\mathcal{C}})$ communication triples in $\mathcal{R}$. Moreover, $\mathcal{R}$ contains at least one $*$-triple because it is canonical. Hence $|\mathcal{R}| \geq \mathrm{fas}(D_{\mathcal{C}}) + 1$. Therefore, $\mathrm{rmin}^*(\mathcal{C}) \geq \mathrm{fas}(D_{\mathcal{C}}) + 1$.

Let $\mathcal{C}_0$ be the set of communication triples corresponding to a minimum feedback arc set F of $D_{\mathcal{C}}$. Set $\mathcal{C}' = \mathcal{C} \setminus \mathcal{C}_0$. The digraph $D_{\mathcal{C}}'$ is $D_{\mathcal{C}} \setminus F$ and so is acyclic. Therefore, by Lemma 7, there is a list $\mathcal{R}'$ of size at most $n + m - 1$ emulating $\mathcal{C}'$ with no global triple. Now the list $\mathcal{C}_0, \mathcal{R}'$ has size at most $\mathrm{fas}(D_{\mathcal{C}}) + n + m - 1$ and emulates $\mathcal{R}$. Hence $\mathrm{rmin}(\mathcal{C}) \leq \mathrm{fas}(D_{\mathcal{C}}) + n + m - 1$. $\qquad\square$

Lemma 8 *If $\mathcal{C}$ is a full set of communication triples with n sources and m destinations, then $\mathrm{rmin}(\mathcal{C}) \geq \mathrm{fas}(D_{\mathcal{C}}) + \min\{n, m\}$.*

Proof Let $\mathcal{R}$ be a canonical routing list with no global triple. A block routes either communications with a same source or communications with a same destination, thus the number of communications routed by a block is at most $\max\{n, m\}$. Since the set of communications is full, there are nm communications. Hence, there must be at least $nm/\max\{n, m\} = \min\{n, m\}$ blocks. Each block contains a $*$-triple, so they are at least $\min\{n, m\}$ $*$-triples in $\mathcal{R}$. Now, by Lemma 6, there are at least $\mathrm{fas}(D_{\mathcal{C}})$ communication triples in $\mathcal{R}$. Hence $\mathcal{R}$ is of size at least $\mathrm{fas}(D_{\mathcal{C}}) + \min\{n, m\}$. $\qquad\square$

4 Complexity results

Assume that $\mathcal{C}$ has a unique port p in $\mathcal{C}$. Then $\mathrm{rmin}^*(\mathcal{C}) = 1$ because the list $((*, *, p))$ emulates $\mathcal{C}$, and so 1-Port Routing List is trivially constant-time solvable.

4.1 Solving 1-Port Without Global Routing List in polynomial time

Theorem 2 1-Port Without Global Routing List *can be solved in polynomial time.*

Proof Let $\mathcal{C}$ be a set of communication triples, all having the same port p. Let S and T be the set of sources and destinations, respectively, of $\mathcal{C}$.

Observe that a routing list (with no global triples) emulating $\mathcal{C}$ can always be transformed into a routing list with no communication triples. Indeed each communication triple (s, t, p) can be replaced by the source triple $(s, *, p)$ (or the destination triple $(*, t, p)$). Therefore, we may only search for a shortest routing list with no communication triples. Such a list is a set $\mathcal{R}$ of source triples and destination triples such that, for every triple (s, t, p) of $\mathcal{C}$, either $(s, *, p) \in \mathcal{R}$ or $(*, t, p) \in \mathcal{R}$. This corresponds to finding a minimum vertex cover in the bipartite graph G defined as follows:

- $V(G) = S \cup T$;
- $E(G) = \{st \mid (s, t, p) \in \mathcal{C}\}$.

Recall that a *vertex cover* in G is a subset W of $V(G)$ such that each edge $e \in E(G)$ has an endvertex in W. Let $\rho(G)$ be the size of a minimum vertex cover in G. A well-known theorem of Gallai [5] asserts that $|V(G)| = \rho(G) + \mu(G)$, where $\mu(G)$ is the size of a maximum matching in G. Now, finding a maximum matching in a bipartite graph can be done in polynomial time, by the Hungarian Method for example. Hence 1-Port Without Global Routing List can be solved in polynomial time. $\qquad\square$

4.2 NP-hardness of 2-Port Routing List and 2-Port Without-global Routing List

Recall that a communication set is *full* if it is of the form $S \times T$, and that a set of communication triples is *full* if it is a set of communication triples on a full set of communications. A *bipartite tournament* is a bipartite digraph D with bipartition (A, B) such that for every pair (a, b), exactly one of the arcs ab and ba is in D. Observe that $\mathcal{C}$ is a full set of communication triples if and only if $D_\mathcal{C}$ is a bipartite tournament.

Theorem 3 2-Port Without-global Routing List *is NP-complete even if $\mathcal{C}$ is restricted to be a full set of communication triples.*

Proof Reduction from Feedback Arc Set on balanced bipartite tournaments which is NP-complete according to Lemma 5. Let D be a balanced bipartite tournament with bipartition (S, T) and let $n = |S| = |T|$. Let $\mathcal{C}$ be the full set of communication triples such that $\mathcal{D}_\mathcal{C} = D$: its source set is S, its destination set is T and $(s, t, p_1) \in \mathcal{C}$ if $st \in A(D)$ and $(s, t, p_2) \in \mathcal{C}$ if $ts \in A(D)$. Let S' (resp. T') be a set of $2n^3$ sources (resp. destinations) disjoint from S (resp. T), and let (T'_1, T'_2) be a balanced partition of T' (i.e. $|T'_1| = |T'_2| = n^3$). Let $\mathcal{C}'$ be the full set of communication triples defined by

$$\mathcal{C}' = \mathcal{C} \cup \{(s, t', p_1) \mid s \in S, t' \in T'\} \cup \{(s', t, p_2) \mid s' \in S', t \in T\}$$
$$\cup \{(s', t', p_1) \mid s' \in S', t' \in T'_1\} \cup \{(s', t', p_2) \mid s' \in S', t' \in T'_2\}.$$

Observe that the list $\mathcal{R}'$ made of all communication triples of $\mathcal{C}$ followed by all $(s, *, p_1)$, for $s \in S$, then all $(*, t', p_2)$ for $t' \in T \cup T'_2$, and finally all $(*, t', p_1)$ for $t' \in T'_1$ emulates $\mathcal{C}'$. Moreover $|\mathcal{R}'| = 2n^3 + 2n + n^2$.

Let $\mathcal{R}$ be shortest canonical routing list with no global triple emulating $\mathcal{C}'$. By definition, $|\mathcal{R}| \leq 2n^3 + 2n + n^2$. We now prove some more properties of $\mathcal{R}$.

(a) $\mathcal{R}$ *contains either a t'-destination triple for all $t' \in T'$, or an s'-source triple for all $s' \in S'$.*

Indeed, if it were not the case, then there is $s' \in S'$ and $t' \in T'$ such that there is no s'-source triple and no t'-destination triple. Let $\mathcal{R}_{s'}$ be the set of triples routing a communication triple of $\mathcal{C}'$ with source s'. Since there is no source triple in $\mathcal{R}_{s'}$, the $2n^3 + n$ communication triple with source s' are routed by distinct destination triples or communication triples. Hence $|\mathcal{R}_{s'}| = 2n^3 + n$. Similarly, the set $\mathcal{R}_{t'}$ of triples routing a communication triple of $\mathcal{C}'$ with destination t' has cardinality $2n^3 + n$ and contains only source or communication triples. Now, $|\mathcal{R}_{s'} \cap \mathcal{R}_{t'}| \leq 1$. Hence $|\mathcal{R}| \geq 4n^3 + 2n - 1 > 2n^3 + 2n + n^2$, a contradiction. This proves (a).

(b) $\mathcal{R}$ *contains a t'-destination triple for all $t' \in T'$.*

If not, then by (a), $\mathcal{R}$ contains an s'-source triple for all $s' \in S'$. Now let $(s', *, p)$ be the first source triple with source in S' that appears in $\mathcal{R}$. In $\mathcal{C}'$, there are at least n^3 communication triples whose port is not p. All these triples are routed by distinct destination or communication triples. Hence $\mathcal{R}$

contains at least $2n^3$ source triples and at least n^3 other triples. So $|\mathcal{R}| \geq 3n^3 > 2n^3 + 2n + n^2$, a contradiction. This proves (b).

(c) *$\mathcal{R}$ contains the $2n^3$ triples of $\{(*, t', p_1) \mid t' \in T_1'\} \cup \{(*, t', p_2) \mid t' \in T_2'\}$.*

Assume that $\mathcal{R}$ does not contain a triples $(*, t', p_1)$ for some $t' \in T_1'$. Then the triples of $\{(s', t', p_1) \mid s' \in S'\}$ are routed by distinct source or communication triples. Hence $|\mathcal{R}| \geq 4n^3 > 2n^3 + 2n + n^2$, a contradiction. We get a similar contradiction if $\mathcal{R}$ does not contain a triples $(*, t', p_2)$ for some $t' \in T_2'$. This proves (c).

(d) *$\mathcal{R}$ contains $\{(*, t, p_2) \mid t \in T\}$.*

Assume for a contradiction that $\mathcal{R}$ does not contain a triples $(*, t, p_2)$ for some $t \in T$. Then the triples of $\{(s', t, p_2) \mid s' \in S'\}$ are routed by distinct source or communication triples. Hence $|\mathcal{R}| \geq 4n^3 > 2n^3 + 2n + n^2$, a contradiction. This proves (d).

(e) *$\mathcal{R}$ contains $\{(s, *, p_1) \mid s \in S\}$.*

Assume for a contradiction that $\mathcal{R}$ does not contain a triples $(s, *, p_1)$ for some $s \in S$. Then the n^3 triples of $\{(s, t', p_1) \mid t' \in T_2'\}$ must be in $|\mathcal{R}|$. Hence $|\mathcal{R}| \geq 3n^3 > 2n^3 + 2n + n^2$, a contradiction. This proves (e).

(f) *The only $*$-triple in $\mathcal{R}$ are those in $\{(*, t, p_2) \mid t \in T\} \cup \{(s, *, p_1) \mid s \in S\} \cup \{(*, t', p_1) \mid t' \in T_1'\} \cup \{(*, t', p_2) \mid t' \in T_2'\}$. Furthermore, free to permute some blocks, we may assume that the last $2n^3$ triples of $\mathcal{R}$ are those of $\{(*, t', p_1) \mid t' \in T_1'\} \cup \{(*, t', p_2) \mid t' \in T_2'\}$.* Assume for a contradiction, that a triple not in the above mentioned set in $\mathcal{R}$. Since there is at most one $*$-triple by source or destination, it must be a source triple with source in S'. But all communication triples routed by such a triple would be correctly routed by the triples in $\{(*, t, p_2) \mid t \in T\} \cup \{(*, t', p_1) \mid t' \in T_1'\} \cup \{(*, t', p_2) \mid t' \in T_2'\}$. Hence removing a source triple with source in S' would leave a routing list emulating $\mathcal{C}$, a contradiction. This proves the first statement of (f).
Let $\mathcal{B}_1, \ldots, \mathcal{B}_q$ be the blocks of $\mathcal{R}$, and let I be the set of indices i for which the block $\mathcal{B}_i$ terminates with a destination triple in $\{(*, t', p_1) \mid t' \in T_1'\} \cup \{(*, t', p_2) \mid t' \in T_2'\}$. The list consisting of the blocks $\mathcal{B}_i$, for $i \notin I$, in the order of $\mathcal{R}$ (i.e. in increasing order of their index), followed by the blocks $\mathcal{B}_i$, for $i \in I$ in any order, also emulates $\mathcal{C}$. This proves the second statement of (f).

Now the order of the $2n$ first blocks $\mathcal{B}_1, \ldots, \mathcal{B}_{2n}$ of $\mathcal{R}$ corresponds to an ordering $\sigma = (v_1, \ldots, v_{2n})$ of the vertices of D, where v_i is either the source s if the $*$-triple of $\mathcal{B}_i$ is $(s, *, p_1)$ or the destination t if the $*$-triple of $\mathcal{B}_i$ is $(*, t, p_2)$. Consider now the set of communication triples in each block $\mathcal{B}_i$. If v_i is a source, then it must contains all communication triples with source v_i and port p_1, that have not been routed previously. By definition of $D_{\mathcal{C}}$, this corresponds to the set of σ-feedback arcs with head v_i. Similarly, if v_i is a destination, the set of communication triples in $\mathcal{B}_i$ corresponds to the set of σ-feedback arcs with head v_i. Thus $|\mathcal{R}| \leq 2n^3 + 2n + \mathrm{fas}(D)$.

Reciprocally, given an ordering σ of D, one can emulate $\mathcal{C}$ with the routing list of size $2n^3 + 2n + \mathrm{fas}(D)$ obtained as follows. Take the blocks terminating

with the $*$-triples of $\{(s, *, p_1) \mid s \in S\} \cup \{(*, t, p_2) \mid t \in T\}$ in the ordering corresponding to σ, and then use the $2n^3$ triples of $\{(*, t', p_1) \mid t' \in T'_1\} \cup \{(*, t', p_2) \mid t' \in T'_2\}$.

Hence $|\mathcal{R}| = 2n^3 + 2n + \mathrm{fas}(D)$. $\square$

Theorem 4 2-Port Routing List *is NP-complete even if $\mathcal{C}$ is restricted to be a full set of communication triples.*

Proof Reduction from Feedback Arc Set on balanced bipartite tournaments. Let D be a balanced bipartite tournament with bipartition (S, T). Set $n = |S| = |T|$. Let $\mathcal{C}$ be the full set of communication triples such that $\mathcal{D}_{\mathcal{C}} = D$: its source set is S, its destination set is T and $(s, t, p_1) \in \mathcal{C}$ if $st \in A(D)$ and $(s, t, p_2) \in \mathcal{C}$ if $ts \in A(D)$. Let $S' \cup S''$ be a set of $n^4 + n^3$ sources disjoint from S with $|S'| = n^4$ and $|S''| = n^3$, and let T' be a set of n^4 destinations disjoint from T.

Let $\mathcal{C}'$ be the full set of communication triples defined by

$$\mathcal{C}' = \mathcal{C} \cup \{(s, t', p_1) \mid s \in S, t' \in T'\} \cup \{(s', t, p_2) \mid s' \in S', t \in T\}$$
$$\cup \{(s'', t', p_1) \mid s'' \in S'', t' \in T'\} \cup \{(s', t', p_2) \mid s' \in S', t' \in T'\}.$$

Observe that the list $\mathcal{R}'$ made of all communication triples of $\mathcal{C}$ followed by all $(*, t, p_2)$ for $t \in T$, then all $(s'', *, p_1)$, for $s'' \in S \cup S''$, and finally $(*, *, p_2)$ emulates $\mathcal{C}'$. Moreover $|\mathcal{R}'| = n^3 + n^2 + 2n + 1$.

Let $\mathcal{R}$ be shortest canonical routing list with global triple emulating $\mathcal{C}'$. By definition, $|\mathcal{R}| \le n^3 + n^2 + 2n + 1$. We now prove some more properties of $\mathcal{R}$.

(a) $\mathcal{R}$ *contains* $(*, *, p_2)$

Assume not. Then the communication triples with source in S' and destination in T' require at least n^4 triples to be routed. Hence $|\mathcal{R}| \ge n^4 > n^3 + n^2 + 2n + 1$, a contradiction. This proves (a).

(b) $\mathcal{R}$ *contains* $(s'', *, p_1)$ *for all* $s'' \in S \cup S''$.

Assume for a contradiction that $\mathcal{R}$ does not contain $(s'', *, p_1)$ for some $s'' \in S \cup S''$. Then the n^4 communication triples (s'', t', p_1), $t' \in T'$, must be routed by distinct communication or destination triples. Thus $|\mathcal{R}| \ge n^4 > n^3 + n^2 + 2n + 1$, a contradiction. This proves (b).

(c) $\mathcal{R}$ *contains* $(*, t, p_2)$ *for all* $t \in T$.

Assume for a contradiction that $\mathcal{R}$ does not contain $(*, t, p_2)$ for some $t \in T$. Then the n^3 communication triples (s'', t, p_2), $s'' \in S''$ must be in $\mathcal{R}$ because $\mathcal{R}$ contains $(s'', *, p_1)$ and thus not $(s'', *, p_2)$. Hence with the n^3 source triples with source in S'', we obtain $|\mathcal{R}| \ge 2n^3 > n^3 + n^2 + 2n + 1$, a contradiction. This proves (c).

(d)   *The only $*$-triple in $\mathcal{R}$ are those in* $\mathcal{F} = \{(s'', *, p_1) \mid s'' \in S \cup S''\} \cup \{(*, t, p_2) \mid t \in T\} \cup \{(*, *, p_2)\}$. *Moreover, free to permute blocks of $\mathcal{R}$, we may assume that the last $n^3 + 1$ triples of $\mathcal{R}$ are the $(s'', *, p_1)$ for all $s'' \in S''$ (in any order) followed by $(*, *, p_2)$.*

Assume for a contradiction that $\mathcal{R}$ is a $*$-triple τ not in $\mathcal{F}$. Since $\mathcal{R}$ has at most one $*$-triple by source or destination, then τ is either a source triple with source in S' or a destination triple with destination in T'. Hence, if we remove τ, all triples routed by τ will be correctly routed by the triples in $\mathcal{F}$. Hence $\mathcal{R} \setminus \tau$ also emulates $\mathcal{R}$, a contradiction to the minimality of $\mathcal{R}$. This proves the first statement of (d).

Let $\mathcal{B}_1, \ldots, \mathcal{B}_q$ be the blocks of $\mathcal{R}$, and let I be the set of indices i for which the block $\mathcal{B}_i$ terminates with a destination triple in $\{(s'', *, p_1) \mid s'' \in S''\} \cup \{(*, *, p_2)\}$. The list consisting of the blocks $\mathcal{B}_i$, for $i \notin I$, in the order of $\mathcal{R}$ (i.e. in increasing order of their index), followed by the blocks $\mathcal{B}_i$, for $i \in I$ in any order, also emulates $\mathcal{C}$. This proves the second statement of (d).

In the exact same way as the end of the proof of Theorem 3, one shows that $|\mathcal{R}| = n^3 + 2n + 1 + \mathrm{fas}(D)$. $\qquad\square$

5 Approximation algorithms

5.1 Heuristics

Let $\mathcal{C}$ be a set of communication triples with source set S and destination set T. Set $n = |S|$ and $m = |T|$. For any port p, let $\mathcal{C}(p)$ be the set of triples of $\mathcal{C}$ with port p. For a source s (resp. destination t) and a port p, let $\mathcal{C}(s, p)$ (resp. $\mathcal{C}(s, t)$) be the set of triples of $\mathcal{C}$ with source s (resp. destination t) and port p. For every source s, let $M(s) := \max_p |\mathcal{C}(s, p)|$ be the maximum number of triples in $\mathcal{C}$ with source s and same port, and for every destination t, let $M(t) := \max_p |\mathcal{C}(t, p)|$ be the maximum number of triples in $\mathcal{C}$ with destination t and same port. Set

$$Z^-(\mathcal{C}) = \sum_{s \in S}(M(s) - 1) = \sum_{s \in S} M(s) - n \quad \text{and}$$

$$Z^|(\mathcal{C}) = \sum_{t \in T}(M(t) - 1) = \sum_{t \in T} M(t) - m.$$

Any routing list with no global triple emulating $\mathcal{C}$ yields an upper bound on $\mathrm{rmin}(\mathcal{C})$. One such list can be obtained by compressing source by source. One source s after another we emulate all triples of $\mathcal{C}$ with source s. This can be done by using the triple $(s, *, p)$ for p a port such that there are $M(s)$ triples with source s and port p after all triples with source s and port distinct from p. Doing so, we save $M(s) - 1$ triples when emulating the triples with source s. Hence, we obtain a routing list of size $|\mathcal{C}| - Z^-(\mathcal{C})$. Such a list is called a *source-based routing list*.

Proceeding similarly according to the destinations, we obtain a routing list, called *destination-based* of size $|\mathcal{C}| - Z^|(\mathcal{C})$.

Setting $Z(\mathcal{C}) = \max\{Z^-(\mathcal{C}), Z^|(\mathcal{C})\}$, we have

$$\mathrm{sav}(\mathcal{C}) \geq Z(\mathcal{C}) \text{ and } \mathrm{rmin}(\mathcal{C}) \leq |\mathcal{C}| - Z(\mathcal{C}). \tag{1}$$

The algorithm consisting in computing a source-based routing list and a destination-based routing list and taking the shortest of the two, is called the *Direction-based Heuristic*. It provides a routing list emulating $\mathcal{C}$ of size $|\mathcal{C}| - Z(\mathcal{C})$. As we shall see in Corollary 2, the Direction-based Heuristic is a 2-approximation for WITHOUT-GLOBAL LIST REDUCTION.

Similarly, any routing list emulating $\mathcal{C}$, possibly with a global triple, yields an upper bound on $\mathrm{rmin}^*(\mathcal{C})$. One can easily obtain such a list as follows. We find a port p such that $|\mathcal{C}(p)|$ is maximum, and we replace the elements of $C(p)$ by the global triple $(*, *, p)$. Such a list is called a *default-port routing list*.

The algorithm consisting in computing a source-based routing list, a destination-based routing list, and a default-port routing list and taking the shortest of the three, is called the *Global Heuristic*. As we shall see in Corollary 3, the Global Heuristic is a 3-approximation for LIST REDUCTION.

5.2 Relation between the problems with and without global triple

Lemma 9 *If* rmin^* *can be approximated with ratio* α *in polynomial time, then* rmin *can be approximated with ratio* $\alpha + 1$ *in polynomial time.*

Proof Assume that there is an algorithm $\mathbb{A}^*$, that given a set of communication triples $\mathcal{C}$, returns a list $\mathcal{R}^*$ emulating $\mathcal{C}$ possibly with a global triple such that $|\mathcal{R}^*| \leq \alpha \, \mathrm{rmin}^*(\mathcal{C})$.

Let $\mathbb{A}$ be the algorithm that does the following: It first runs $\mathbb{A}^*$ to get a list $\mathcal{R}^*$ emulating $\mathcal{C}$ such that $|\mathcal{R}^*| \leq \alpha \, \mathrm{rmin}^*(\mathcal{C})$. If $\mathcal{R}^*$ contains no global triple, then it returns $\mathcal{R} := \mathcal{R}^*$. Then $|\mathcal{R}| \leq \alpha \, \mathrm{rmin}^*(\mathcal{C}) \leq \alpha \, \mathrm{rmin}(\mathcal{C})$. Assume now that $\mathcal{R}^*$ contains a global triple. Free to delete, the useless triples after the first global triple, we may assume that $\mathcal{R}^*$ has a unique global triple, and that it is the final one. Let $\tau = (*, *, p)$ be this triple, and let $\mathcal{C}_\tau$ be the set of triples of $\mathcal{C}$ that are routed by it when routing according to $\mathcal{R}^*$. By definition, all the triples of $\mathcal{C}_\tau$ have port p. Therefore, by Theorem 2, one can find in polynomial time a shortest routing list $\mathcal{R}_\tau$ emulating $\mathcal{C}_\tau$. We then return the concatenation $\mathcal{R}$ of $\mathcal{R}^* \setminus \tau$ and $\mathcal{R}_\tau$. Now $|\mathcal{R}| \leq |\mathcal{R}^*| + |\mathcal{R}_\tau| \leq \alpha \, \mathrm{rmin}^*(\mathcal{C}) + \mathrm{rmin}(\mathcal{C}_\tau) - 1$. But $\mathrm{rmin}^*(\mathcal{C}) \leq \mathrm{rmin}(\mathcal{C})$ and $\mathrm{rmin}(\mathcal{C}_\tau) \leq \mathrm{rmin}(\mathcal{C})$ because $\mathcal{C}_\tau \subseteq \mathcal{C}$. Hence $|\mathcal{R}| \leq (\alpha + 1) \, \mathrm{rmin}(\mathcal{C}) - 1$.   □

Lemma 10 *If* sav *can be approximated with ratio* α *in polynomial time, then* sav^* *can be approximated with ratio* $\alpha + 1$ *in polynomial time.*

Proof Assume that there is an algorithm $\mathbb{A}$, that given a set of communication triples $\mathcal{C}$, returns a list $\mathcal{R}$ emulating $\mathcal{C}$ with no global triple such that $\mathrm{sav}(\mathcal{C}) \leq \alpha(|\mathcal{C}| - |\mathcal{R}|)$.

Let $\mathbb{A}^*$ be the algorithm that does the following:

1. It first runs $\mathbb{A}$ to get a list $\mathcal{R}$ emulating $\mathcal{C}$ such that $\mathrm{sav}(\mathcal{C}) \leq \alpha(|\mathcal{C}| - |\mathcal{R}|)$.
2. Next, it computes a default-port routing list $\mathcal{R}'$.
3. It returns the list $\mathcal{R}^*$ which is the shortest among $\mathcal{R}$ and $\mathcal{R}'$.

Since $\mathbb{A}$ runs in polynomial time, so does $\mathbb{A}^*$, because its steps 2 and 3 above can trivially be performed in polynomial time. Let us now prove that $\mathbb{A}^*$ gives an $(\alpha + 1)$-approximate solution, that is $\mathrm{sav}^*(\mathcal{C}) \leq (\alpha + 1)(|\mathcal{C}| - |\mathcal{R}^*|)$. By Lemma 2, we have $\mathrm{sav}^*(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C}) + M(\mathcal{C}) - 1$. But $\mathrm{sav}(\mathcal{C}) \leq \alpha(|\mathcal{C}| - |\mathcal{R}|) \leq \alpha(|\mathcal{C}| - |\mathcal{R}^*|)$, and by definition, $|\mathcal{C}| - |\mathcal{R}^*| \geq |\mathcal{C}| - |\mathcal{R}'| = M(\mathcal{C}) - 1$. It follows that $\mathrm{sav}^*(\mathcal{C}) \leq (\alpha + 1)(|\mathcal{C}| - |\mathcal{R}^*|)$. $\qquad\square$

5.3 Approximate upper bounds on sav and sav*

Theorem 5 *Let $\mathcal{C}$ be a set of communication triples. Then*

$$Z(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C}) \leq 2Z(\mathcal{C}).$$

Proof By Equation (1), we have $Z(\mathcal{C}) \leq \mathrm{sav}(\mathcal{C})$.

Let us now prove $W(\mathcal{C}) \leq 2Z(\mathcal{C})$. Let $\mathcal{R}$ be a shortest routing list with no global triple emulating $\mathcal{C}$. Let S' (resp. T') be the set of sources (resp. destinations) x such that $\mathcal{R}$ contains a $*$-triple with source (resp. destination) x. For each $x \in S' \cup T'$, let $P(x)$ be the number of triples of $\mathcal{C}$ that are routed by τ_x^*, the $*$-triple with source or destination x. Clearly, $\mathrm{sav}(\mathcal{C}) = |\mathcal{C}| - |\mathcal{R}| = \sum_{x \in S' \cup T'}(P(x) - 1)$. Moreover $P(x) \leq M(x)$ for every x because all communication triples routed by τ_x^* have the same port (the one of τ_x^*). Therefore

$$\begin{aligned}
\mathrm{sav}(\mathcal{C}) &\leq \sum_{x \in S' \cup T'}(P(x) - 1) \leq \sum_{x \in S' \cup T'}(M(x) - 1) \\
&\leq \sum_{s \in S}(M(s) - 1) + \sum_{t \in T}(M(t) - 1) \\
&\leq Z^-(\mathcal{C}) + Z^|(\mathcal{C}) \leq 2Z(\mathcal{C})
\end{aligned} \qquad (2)$$

$\qquad\square$

The source-based and destination-based routing lists can obviously be computed in polynomial time.

Corollary 2 *The Direction-based Heuristic is a 2-approximation for* WITHOUT-GLOBAL LIST REDUCTION*.*

The ratio 2 is tight for the Direction-based Heuristic.

Proposition 1 *There are sets of communication triples $\mathcal{C}$ such that the list returned by the Direction-based Heuristic has size $|\mathcal{C}| - \frac{1}{2}\,\mathrm{sav}(\mathcal{C})$.*

Proof Consider the following set of communication triples:

$$\mathcal{C} = \{(s_1, t_j, p_1) \mid 2 \leq j \leq \ell + 2\} \cup \{(s_i, t_1, p_2) \mid 2 \leq i \leq \ell + 2\}.$$

One can easily check that $Z^-(\mathcal{C}) = Z^|(\mathcal{C}) = Z(\mathcal{C}) = \ell$ while $\mathrm{sav}(\mathcal{C}) = 2\ell$. $\qquad\square$

In Equation (4), we use the inequality $Z^-(\mathcal{C}) + Z^|(\mathcal{C}) \le 2Z(\mathcal{C})$ to obtain that the Direction-based Heuristic is a 2-approximation. But if we do not use this inequality, we obtain that the Direction-based Heuristic is a $\frac{Z^-(\mathcal{C})+Z^|(\mathcal{C})}{Z(\mathcal{C})}$-approximation. Observe that the ratio $\frac{Z^-(\mathcal{C})+Z^|(\mathcal{C})}{Z(\mathcal{C})}$ can be computed while running the Direction-based heuristic and is often smaller than 2.

We believe that more elaborate heuristics for WITHOUT-GLOBAL LIST REDUCTION can reach better approximation ratio than 2.

Problem 1 What is the best approximation ratio for WITHOUT-GLOBAL LIST REDUCTION?

Together with Lemma 10, Corollary 2 immediately yields the following.

Corollary 3 *The Global Heuristic is a 3-approximation for* LIST REDUCTION.

The ratio 3 is tight for the Global Heuristic.

Proposition 2 *There are sets of communication triples $\mathcal{C}$ such that the list returned by the Global Heuristic has size $|\mathcal{C}| - \frac{1}{3}\operatorname{sav}(\mathcal{C})$.*

Proof Let $\gamma(i,j)$ be the integer in $\{1, \ldots, \ell+1\}$ congruent to $(i-1)$ modulo $\ell+1$. Consider the following set of communication triples:

$$\mathcal{C} = \{(s_i, t_j, p_{\gamma(i,j)}) \mid 1 \le i \le \ell+1 \text{ and } 1 \le j \le \ell+1\} \cup$$
$$\{(s_{\ell+2}, t_j, p_{\ell+2}) \mid 1 \le j \le \ell+1\} \cup \{(s_i, t_{\ell+2}, p_{\ell+3}) \mid 1 \le i \le \ell+1\}$$

Observe that each port appears exactly $\ell+1$ times. Consequently, the default-port routing list has size $|\mathcal{C}| - \ell$. Observe moreover that $M(s_i) = 1$ for $1 \le i \le \ell+1$ and $M(s_{\ell+2} = \ell+1)$, and $M(t_j) = 1$ for $1 \le j \le \ell+1$ and $M(t_{\ell+2} = \ell+1)$. Therefore $Z(\mathcal{C}) = \ell$. Hence the list returned by the Global Heuristic has size $|\mathcal{C}| - \ell$.

But $sav^*(\mathcal{C}) = 3\ell$, because using the triples $(s_{\ell+2}, *p_{\ell+2})$, $(*, t_{\ell+2}, p_{\ell+3})$, and $(*, *, p_1)$ one can save 3ℓ communication triples. $\square$

We believe however that the approximation of 3 is not best possible for LIST REDUCTION.

Problem 2 What is the best approximation ratio for LIST REDUCTION?

5.4 Generalization to higher dimensions

We show here that similar approximation algorithms can be defined for higher dimension matrices. This corresponds to a routing based on more than two fields (e.g. on three fields, source address, destination address and protocol).

Consider a model with f fields. A routing list is now an ordered list of $(f+1)$-uples $(e_1, ..., e_f, p)$. Let $\mathcal{C}$ be a set of communication $(f+1)$-uples with field set F_j for $1 \le j \le f$.

In this case, the aggregation can be done on any subset of fields. Let $\sigma = \{j_1, \ldots, j_g\}$ be a subset of fields. Set $F^\sigma = F_{j_1} \times \cdots \times F_{j_g}$. For every element $\mathbf{e} = (e_{j_1} \ldots, e_{j_g}) \in F^\sigma$, let $\mathcal{C}^\sigma(\mathbf{e}, p)$ be the set of $(f + 1)$-uples of $\mathcal{C}$ whose j_ith element is e_{j_i} for all $1 \leq i \leq g$ and whose port is p. Furthermore, let $M^\sigma(\mathbf{e}) := \max_p |\mathcal{C}(\mathbf{e}, p)|$ be the maximum number of $(f + 1)$-uples in $\mathcal{C}$ whose j_ith element is e_{j_i} for all $1 \leq i \leq g$ and whose port is p. Set

$$Z^\sigma(\mathcal{C}) = \sum_{\mathbf{e} \in F^\sigma} (M^\sigma(\mathbf{e}) - 1) = \sum_{\mathbf{e} \in F^\sigma} M^\sigma(\mathbf{e}) - |F^\sigma|.$$

For every element $\mathbf{e} = (e_{j_1} \ldots, e_{j_g}) \in F^\sigma$, we denote by $(\mathbf{e}, *, p)^\sigma$ the $(f+1)$-uple whose j_ith element is e_{j_i} for all $1 \leq i \leq g$, whose jth element is $*$ for all $j \notin \sigma$, and whose port is p. For every $\mathbf{e} \in F^\sigma$, we use the $(f+1)$-tuple $(\mathbf{e}, *, p)^\sigma$ for p a port such that there are $M^\sigma(\mathbf{e})$ $(f + 1)$-uples whose j_ith element is e_{j_i} for all $1 \leq i \leq g$ and whose port is p, after all $(f + 1)$-uples whose j_ith element is e_{j_i} for all $1 \leq i \leq g$ and whose port is distinct from p. We obtain a routing list of size $|\mathcal{C}| - Z^\sigma(\mathcal{C})$. Such a list is called a σ-*based routing list*. Note that, in the case when $f = 2$, the $\{1\}$-based routing list is the source-based routing list, the $\{2\}$-based routing list is the destination-based routing list, and the $\{1, 2\}$-based routing list is the default-port routing list.

Setting $Z^*(\mathcal{C}) = \max\{Z^\sigma(\mathcal{C}) \mid \sigma \subseteq \{1, \ldots, f\}, \sigma \neq \emptyset\}$, we have

$$\text{sav}^*(\mathcal{C}) \geq Z^*(\mathcal{C}) \text{ and } \text{rmin}(\mathcal{C}) \leq |\mathcal{C}| - Z^*(\mathcal{C}). \tag{3}$$

The algorithm consisting in computing the σ-based routing lists for all non-empty subset σ of $\{1, \ldots, f\}$ (there are $2^f - 1$ such σ), and taking the shortest of the lists, is called the *General Heuristic (for f fields)*. It provides a routing list emulating $\mathcal{C}$ of size $|\mathcal{C}| - Z^*(\mathcal{C})$. As we now prove this heuristic is a $(2^f - 1)$-approximation for f-Field List Reduction.

Theorem 6 *Let $\mathcal{C}$ be a set of communication $(f + 1)$-uples. Then*

$$Z^*(\mathcal{C}) \leq \text{sav}^*(\mathcal{C}) \leq (2^f - 1)Z^*(\mathcal{C}).$$

Proof The proof is similar to the one with two fields. By Equation (3), we have $Z^*(\mathcal{C}) \leq \text{sav}^*(\mathcal{C})$.

Let us now prove $\text{sav}^*(\mathcal{C}) \leq (2^f - 1)Z^*(\mathcal{C})$. Let $\mathcal{R}$ be a shortest routing list emulating $\mathcal{C}$. Let E^σ be the set of elements $\mathbf{e} \in F^\sigma$ such that $\mathcal{R}$ contains a $(f + 1)$-uple $\tau_\mathbf{e}^*$ equal to $(\mathbf{e}, *, p)^\sigma$ for some port p.

For each $\mathbf{e} \in F^\sigma$, let $P^\sigma(\mathbf{e})$ be the number of communication $(f+1)$-uples of $\mathcal{C}$ that are routed by $\tau_\mathbf{e}^*$. Clearly, $\text{sav}^*(\mathcal{C}) = |\mathcal{C}| - |\mathcal{R}| = \sum_\sigma \sum_{\mathbf{e} \in F^\sigma} (P^\sigma(\mathbf{e}) - 1)$. Moreover $P^\sigma(\mathbf{e}) \leq M^\sigma(\mathbf{e})$ for every $\mathbf{e} \in F^\sigma$ because all communication $(f+1)$-uples routed by $\tau_\mathbf{e}^*$ have the same port (the one of $\tau_\mathbf{e}^*$). Therefore

$$\text{sav}^*(\mathcal{C}) \leq \sum_\sigma \sum_{\mathbf{e} \in F^\sigma} (P^\sigma(\mathbf{e}) - 1) \leq \sum_\sigma \sum_{\mathbf{e} \in F^\sigma} (M^\sigma(\mathbf{e}) - 1)$$
$$\leq \sum_\sigma Z^\sigma(\mathcal{C}) \leq (2^f - 1)Z^*(\mathcal{C}) \tag{4}$$

$\square$

Each σ-based routing list can obviously be computed in polynomial time.

Corollary 4 *In the problem with f fields, $f \geq 2$, the General Heuristic is a $(2^f - 1)$-approximation for f-FIELD LIST REDUCTION.*

Remark 1 The ratio $2^f - 1$ in Theorem 6 and Corollary 4 comes from the fact that there are $2^f - 1$ non-empty subsets of $\{1, \ldots, f\}$. We can get better approximation ratio if the set of possible $*$-$(f + 1)$-uples is restricted. For example, if we only allow to have at most one $*$ per $(f + 1)$-uple, (as we did in WITHOUT-GLOBAL LIST REDUCTION), then we consider only f possible subsets σ, and so we get an f-approximation for this variant.

5.5 Improving further

We shall now give a parameter $W(\mathcal{C})$ that can be computed in polynomial time, and which is a better upper bound on $\text{sav}(\mathcal{C})$ than $Z^-(\mathcal{C}) + Z^|(\mathcal{C})$. This parameter is interesting because it often provides a better estimate on how approximate is the solution returned by the Direction-based Heuristic. It is also useful because one can sometimes estimate $W(\mathcal{C})$ for some general cases $\mathcal{C}$ and thus obtain good lower bounds on rmin. An example is given in Theorem 8. For sake of brevity, we do not provide the proofs here. The interested reader is referred to [6].

Let $\mathcal{C}$ be a set of communication triples with destination set S and destination set T. Set $n = |S|$ and $m = |T|$. We can order S and T by decreasing order according to the function M. That is $M(s_1) \geq M(s_2) \geq \cdots \geq M(s_n)$ and $M(t_1) \geq M(t_2) \geq \cdots \geq M(t_m)$.

The directed $\{0, 1, \ldots, n\} \times \{0, 1, \ldots, m\}$-grid, denoted by $G_{n,m}$ is the digraph defined by

$$
\begin{aligned}
V(G_{n,m}) &= \{(i, j) \mid 0 \leq i \leq n \text{ and } 0 \leq j \leq m\} \\
A(G_{n,m}) &= \{((i - 1, j), (i, j)) \mid 1 \leq i \leq n \text{ and } 0 \leq j \leq m\} \\
&\quad \cup \{((i, j - 1), (i, j)) \mid 0 \leq i \leq n \text{ and } 1 \leq j \leq m\}.
\end{aligned}
$$

For convenience, we set $a^h(i, j) = ((i - 1, j), (i, j))$ and $a^v(i, j) = ((i, j - 1), (i, j))$.

Let $w_{\mathcal{C}}$ be the weight function defined on $A(G_{n,m})$ by

$$w_{\mathcal{C}}(a^h(i, j)) = \min\{M(s_i), m - j\} - 1 \text{ and } w_{\mathcal{C}}(a^v(i, j)) = \min\{M(t_j), n - i\} - 1$$

and let $W(\mathcal{C})$ be the maximum length of a path in $(G_{n,m}, w_{\mathcal{C}})$. (The *length* of a path is the sum of the weights of its arcs.) Observe that it is attained by a path from $(0, 0)$ to one of the sides $\{n\} \times \{0, 1, \ldots, n\}$ and $\{0, 1, \ldots, n\} \times \{n\}$ because the weight of an arc is negative if and only if it is in $\{a^h(i, m) \mid 1 \leq i \leq n\} \cup \{a^v(n, j) \mid 1 \leq j \leq m\}$. Therefore $W(\mathcal{C})$ can be computed in polynomial time because the longest path from a vertex u to a set X of vertices in a edge-weighted acyclic digraph is polynomial-time computable.

One can show the following theorem.

Theorem 7 $\mathrm{sav}(\mathcal{C}) \leq W(\mathcal{C}) \leq Z^-(\mathcal{C}) + Z^|(\mathcal{C}) \leq 2Z(\mathcal{C})$, *for all set of communication triples* $\mathcal{C}$.

Estimating $W(\mathcal{C})$ for a full set of communication triples, one shows the following theorem.

Theorem 8 *Let* S *be a set of* n *sources,* T *be a set of* n *destinations. Let* $\mathcal{C}$ *be a set of communication triples for* $S \times T$. *If for every source* s, *at most* M *communications with source* s *are assigned the same port, and for every destination* t, *at most* M *of the communications with destination* t *are assigned the same port, then* $\mathrm{rmin}(\mathcal{C}) \geq (n-M)^2 + 2n - M$ *and* $\mathrm{rmin}^*(\mathcal{C}) \geq (n-M)^2 + n - M + 1$.

The lower bound given by Theorem 8 is better than the one given by estimating $Z(\mathcal{C})$. Indeed, for a set $\mathcal{C}$ as in the above theorem, we have $Z^-(\mathcal{C}) = Z^|(\mathcal{C}) = Z(\mathcal{C}) = n(M-1)$, which yields the lower bound $n^2 - 2nM + 2n$ for $\mathrm{rmin}(\mathcal{C})$ which is lower than $(n-M)^2 + 2n - M$. The improvement is particularly dramatic when $M = n/2$. The lower bound on $\mathrm{rmin}(\mathcal{C})$ given by $Z^-(\mathcal{C}) + Z^|(\mathcal{C})$ is $2n$ while the one given by Theorem 8 is $\frac{n^2}{4} + \frac{3n}{2}$.

For every integers M and n such that $M \leq n$, the bound given by Theorem 8 is tight as there exists a full set $\mathcal{C}$ of triples with n sources and n destinations such that

(i) for every source s, at most M communications with source s are assigned the same port,

(ii) for every destination t, at most M communications with destination t are assigned the same port, and

(iii) $\mathrm{rmin}(\mathcal{C}) = (n-M)^2 + 2n - M$.

5.6 Approximating 2-PORT WITHOUT-GLOBAL ROUTING LIST using approximation algorithms for FEEDBACK ARC SET

Van Zuylen [26] gave a 4-approximate polynomial-time algorithm for FEEDBACK ARC SET in complete bipartite tournaments. We can use this approximation algorithm for FEEDBACK ARC SET to build polynomial-time approximation algorithms for some sets of communication triples.

Theorem 9 *There is a polyomial-time 4-approximate algorithm for* 2-PORT WITHOUT-GLOBAL ROUTING LIST *restricted to the full sets of communication triples with* n *sources and* m *destinations such that* $n + m - 1 \leq 4 \cdot \min\{n, m\}$.

Proof Let $\mathcal{C}$ be a set of communication triples with two ports, n sources and m destinations such that $n + m - 1 \leq 4 \cdot \min\{n, m\}$.

The algorithm proceeds as follows. Using Van Zuylen's algorithm, it finds a feedback arc set F of $D_{\mathcal{C}}$ of size at most $4\mathrm{fas}(D_{\mathcal{C}})$. Then, as in the proof of Lemma 7, it derives a routing list $\mathcal{R}$ with no global triple of size at most $4\mathrm{fas}(D_{\mathcal{C}}) + n + m - 1$ which is at most $4\mathrm{fas}(D_{\mathcal{C}}) + 4 \cdot \min\{n, m\}$ by hypothesis.

Now by Lemma 8, $\mathrm{rmin}(\mathcal{C}) \geq \mathrm{fas}(D) + \min\{n, m\}$. Therefore, the size of $\mathcal{R}$ is at most $4 \cdot \mathrm{rmin}(\mathcal{C})$. $\square$

We can also derive approximation algorithms for almost full sets of communication triples. For convenience, we present the result when the number of sources equals the number of destinations. They can easily be extended to the case when those numbers differ.

Theorem 10 *Let α be a positive real number. There is a polyomial-time $\max\{4, 2 + 2\alpha\}$-approximate algorithm for* 2-Port Without-global Routing List *restricted to the sets of at least $n^2 - \alpha \cdot n$ communication triples with n sources and n destinations.*

Proof Let $\mathcal{C}$ be a set of at least $n^2 - \alpha \cdot n$ communication triples with n sources, n destinations and two ports.

Let D_1 (resp. D_2) be the digraph obtained by adding an arc st (resp. ts) for each communication (s, t) not appearing in $\mathcal{C}$. Observe that D_i corresponds to the set of communication triples obtained from $\mathcal{C}$ by routing all missing communications to the port p_i. Both D_1 and D_2 are bipartite tournaments.

Consider a minimum feedback arc set S of $D_{\mathcal{C}}$, and let $\sigma = v_1, \ldots, v_{2n}$ be an ordering such that S is σ-feedback. For each missing communication (s, t) in $\mathcal{C}$, the arc st is *backward* if $s = v_i$, $t = v_j$ and $j < i$, otherwise it is *forward*. Observe that st is forward (resp. backward) if and only if ts is backward (resp. forward). Let $B_1 = \{st \mid st \text{ is backward}\}$ and $B_2 = \{ts \mid st \text{ is backward}\}$. Then for $i = 1, 2$, $S \cup B_i$ is a feedback arc set of D_i. Moreover $|B_1| + |B_2| = \alpha \cdot n$, so $\min\{|B_1|, |B_2|\} \leq \frac{\alpha}{2} \cdot n$. Hence

$$\min\{\mathrm{fas}(D_1), \mathrm{fas}(D_2)\} \leq |S| + \min\{|B_1|, |B_2|\} \leq \mathrm{fas}(D_{\mathcal{C}}) + \frac{\alpha}{2} \cdot n. \quad (5)$$

Now our approximation algorithm proceeds as follows. It constructs D_1 and D_2, compute a feedback arc set of both and takes the smallest one S_m. It then returns the routing list $\mathcal{R}$ consisting of the communication triples corresponding to the arcs of S_m, by adding at most $2n - 1$ appropriate destination or source triples following the proof of Lemma 7.

Now $|R| < |S_m| + 2n \leq 4 \cdot \min\{\mathrm{fas}(D_1), \mathrm{fas}(D_2)\} + 2n \leq 4 \cdot \mathrm{fas}(D_{\mathcal{C}}) + (2\alpha + 2)n$ by Equation (5). But by Lemma 8, $\mathrm{rmin}(\mathcal{C}) \geq \mathrm{fas}(D_{\mathcal{C}}) + n$. Therefore $|R| \leq \max\{4, 2 + 2\alpha\} \cdot \mathrm{rmin}(\mathcal{C})$. $\square$

Problem 3 Can we derive from the fact that Feedback Arc Set is APX-complete, that 2-Port Without-global Routing List is also APX-complete? For which approximation ratio?

The complementary problem to Feedback Arc Set is the following.
Maximum Acyclic Subdigraph:
Input: A digraph D, and an integer k.
Question: Does D have an acyclic subgraph with at least k edges?

The number of edges of an acyclic subdigraph of D is denoted $\mathrm{mas}(D)$. Clearly

$$\mathrm{mas}(D) + \mathrm{fas}(D) = |A(D)| \text{ for all digraph } D.$$

Arora et al. [1] proved that MAXIMUM ACYCLIC SUBDIGRAPH admits a polynomial-time approximation scheme on dense instances. A *dense* digraph is one in which the number of arcs is $\Omega(n^2)$. Hence a natural question is the following:

Problem 4 Does 2-PORT WITHOUT GLOBAL LIST REDUCTION admits a polynomial-time approximation scheme on sets of communication triples whose associated digraph is dense?

5.7 Polynomial-time algorithm when the associated digraph is acyclic

Theorem 11 *There is a polynomial-time algorithm that computes* $\mathrm{rmin}(\mathcal{C})$ *when* $D_{\mathcal{C}}$ *is acyclic.*

The proof of Theorem 11 is based on the following integer linear programming formulation.

$$
\begin{aligned}
\text{Minimize } & \sum x_{(s,t,p)} + \sum x_{(*,t,p)} + \sum x_{(s,*,p)} \\
\text{Subject to: } & x_{(s,t,p)} + x_{(*,t,p)} + x_{(s,*,p)} \geq 1 && \text{for all } (s,t,p) \in \mathcal{C} \\
& \sum_p x_{(*,t,p)} \leq 1 && \text{for all } t \in T \qquad (6) \\
& \sum_p x_{(s,*,p)} \leq 1 && \text{for all } s \in S \\
& x_{(s,t,p)}, x_{(*,t,p)}, x_{(s,*,p)} \in \{0,1\}
\end{aligned}
$$

This translates in the following ILP on $D = \mathcal{D}_{\mathcal{C}}$

$$
\begin{aligned}
\text{Minimize } & \sum_{a \in A(D)} x_a + \sum_{v \in V(D)} (x_v^+ + x_v^-) \\
\text{Subject to: } & x_{uv} + x_u^+ + x_v^- \geq 1 \quad \text{for all } uv \in A(D) \qquad (7) \\
& x_v^+ + x_v^- \leq 1 \quad \text{for all } v \in V(D) \\
& x_a, x_v^+, x_v^- \in \{0,1\}
\end{aligned}
$$

Let us prove that this ILP computes $\mathrm{rmin}(\mathcal{C})$ if $\mathcal{D}_C$ is acyclic.

Proposition 3 *If* $\mathcal{D}_C$ *is acyclic, then the ILP* (6) *computes* $\mathrm{rmin}(\mathcal{C})$.

Proof Let x^* be an optimal solution of the ILP (6) for $\mathcal{C}$ and let r^* be the optimal value of the objective function. Let $\mathcal{R}$ be a shortest routing list (with no global triples) emulating $\mathcal{C}$.

Let $x^{\mathcal{R}}$ be the characteristic function of R, that is $x_\tau^{\mathcal{R}} = 1$ if the triple $\tau \in R$, and $x_\tau^{\mathcal{R}} = 0$ otherwise. For all $(s,t,p) \in \mathcal{C}$, the triple is routed by $\mathcal{R}$ so $x_{(s,t,p)}^{\mathcal{R}} + x_{(*,t,p)}^{\mathcal{R}} + x_{(s,*,p)}^{\mathcal{R}} \geq 1$. Now since R is a shortest list, there is at most one destination triple per destination t, so for all $t \in T$, $\sum_p x_{(*,t,p)}^{\mathcal{R}} \leq 1$. Similarly, there is at most one source triple per source s, so for all $s \in S$, $\sum_p x_{(s,*,p)}^{\mathcal{R}} \leq 1$.

Therefore, $x^{\mathcal{R}}$ satisfies the constraints of (6). Moreover, by definition $|\mathcal{R}| = \sum x^{\mathcal{R}}_{(s,t,p)} + \sum x^{\mathcal{R}}_{(*,t,p)} + \sum x^{\mathcal{R}}_{(s,*,p)}$. Therefore $r^* \leq |\mathcal{R}| = \text{rmin}(\mathcal{C})$.

Reciprocally, let us construct a routing list $\mathcal{R}^*$ emulating $\mathcal{C}$ from x^*. Let $\mathcal{C}'$ be the set of communication triples (s,t,p) such that $x^*_{(s,t,p)} = 1$, let $\mathcal{R}'$ be any list over $\mathcal{C}'$, and set $\mathcal{C}'' = \mathcal{C} \setminus \mathcal{C}'$. Since $D_{\mathcal{C}}$ was acyclic, $D_{\mathcal{C}''}$ is also acyclic. Moreover, the set Z of destination and source triples τ such that $x^*_\tau = 1$ corresponds to a star cover of $D_{\mathcal{C}''}$. Indeed, the first constrainst implies that for every arc uv of $D_{\mathcal{C}''}$, $u \in V^+$ or $v \in V^-$ because $x^*_{(s,t,p)} = 0$, and the second and third constraints imply that two stars have distinct centres. Hence, there is a convenient routing list $\mathcal{R}''$ of size $|Z|$ emulating $\mathcal{C}''$. The concatenation of $\mathcal{R}'$ and $\mathcal{R}''$ is then a routing list emulating $\mathcal{C}$. Hence $\text{rmin}(\mathcal{C}) \leq |\mathcal{R}'| + |\mathcal{R}''| = |\mathcal{C}'| + |\mathcal{C}''| = |\mathcal{C}|$. $\qquad\square$

Lemma 11 *ILP* (6) *can be solved in polynomial time.*

Proof Let $\mathbf{M}_D$ be the matrix associated to ILP *(6)*. Then $\mathbf{M}_D = [\mathbf{I} \ \mathbf{A}_D]$. Since $\mathbf{A}_D$ is totally unimodular, $\mathbf{M}_D$ is also totally unimodular. Thus, by classical results due to Hoffman and Kruskal [11], the associated linear programme has an integral solution (if it has one), and so ILP (6) can be solved in polynomial time $\qquad\square$

Proof (Proof of Theorem 11) It follows directly from Proposition 3 and Lemma 11. $\qquad\square$

6 Conclusion and Open Problems

In this paper, we first provide the hardness of the problem of compressing two dimensional routing tables. We show that the associated decision problems are NP-complete as soon as the number of ports is greater of equal to 2.

We then propose approximation algorithms. In particular, we provide a simple heuristic, called Direction-based Heuristic which is a 2-approximation for WITHOUT-GLOBAL LIST REDUCTION. However, we believe that more elaborate heuristics for WITHOUT-GLOBAL LIST REDUCTION can reach better approximation ratio than 2. We thus leave as open the following problem.

Problem 5 What is the best approximation ratio for WITHOUT-GLOBAL LIST REDUCTION?

We also provide a second heuristic, Global Heuristic, which is a 3-approximation for LIST REDUCTION. Note that this heuristic has been tested for data center scenarios in [20]. Similarly, we believe however that the approximation of 3 is not best possible for LIST REDUCTION.

Problem 6 What is the best approximation ratio for LIST REDUCTION?

We established a strong link between the problems considered in this paper and the FEEDBACK ARC SET problem. It leads us to the following question which we leave as open.

Problem 7 Can we derive from the fact that Feedback Arc Set is APX-complete, that 2-Port Without-global Routing List is also APX-complete? For which approximation ratio?

Arora et al. [1] proved that Maximum Acyclic Subdigraph admits a polynomial-time approximation scheme on dense instances. A *dense* digraph is one in which the number of arcs is $\Omega(n^2)$. Hence a natural question is the following:

Problem 8 Does 2-Port Without Global List Reduction admits a polynomial-time approximation scheme on sets of communication triples whose associated digraph is dense?

Last, we succeeded in generalizing the 3-approximation for List Reduction to a higher number of dimensions to obtain a $(2^f - 1)$-approximation for f-fields. Could we also generalize the 4-approximation for Routing List to higher dimensions?

Problem 9 Does f-Fields Routing List admit a constant ratio approximation algorithm?

References

1. Arora, S., Frieze, A., Kaplan, H.: A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. Mathematical programming **92**(1), 1–36 (2002)
2. Buddhikot, M.M., Suri, S., Waldvogel, M.: Space decomposition techniques for fast layer-4 switching. Springer (2000)
3. Cohen, R., Lewin-Eytan, L., Naor, J., Raz, D.: On the effect of forwarding table size on SDN network utilization. In: IEEE INFOCOM, pp. 1734–1742 (2014)
4. Eppstein, D., Muthukrishnan, S.: Internet packet filter management and rectangle geometry. In: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, pp. 827–835 (2001)
5. Gallai, T.: Maximum-minimum sätze über graphen. Acta Mathematica Hungarica **9**(3), 395–434 (1958)
6. Giroire, F., Havet, F., Moulierac, J.: Compressing Two-dimensional Routing Tables with Order. Research Report RR-8658, INRIA Sophia Antipolis (2014)
7. Giroire, F., Havet, F., Moulierac, J.: Compressing two-dimensional routing tables with order. In: INOC (International Network Optimization Conference). Varsovie, Poland (2015). URL https://hal.inria.fr/hal-01162724
8. Giroire, F., Moulierac, J., Khoa Phan, T.: Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing. In: IEEE GLOBECOM. IEEE, Austin Texas, United States (2014)
9. Guo, J., Hüffner, F., Moser, H.: Feedback arc set in bipartite tournaments is np-complete. Information processing letters **102**(2), 62–65 (2007)
10. Hari, A., Suri, S., Parulkar, G.: Detecting and resolving packet filter conflicts. In: INFOCOM 2000, pp. 1203–1212. IEEE (2000)
11. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: 50 Years of Integer Programming 1958-2008, pp. 49–76. Springer (2010)
12. Kang, N., Liu, Z., Rexford, J., Walker, D.: Optimizing the "one big switch abstraction" in software-defined networks. In: Proceedings of CoNEXT, pp. 13–24. ACM, New York, NY, USA (2013)
13. Kanizo, Y., Hay, D., Keslassy, I.: Palette: Distributing tables in software-defined networks. In: INFOCOM, 2013 Proceedings IEEE, pp. 545–549 (2013)

14. Kann, V.: On the approximability of np-complete optimization problems. Ph.D. thesis, Royal Institute of Technology Stockholm (1992)
15. Karp, R.M.: Reducibility among combinatorial problems. Springer (1972)
16. Kogan, K., Nikolenko, S.I., Rottenstreich, O., Culhane, W., Eugster, P.: Exploiting order independence for scalable and expressive packet classification. IEEE/ACM Transactions on Networking **24**(2), 1251–1264 (2016)
17. Lakshman, T., Stiliadis, D.: High-speed policy-based packet forwarding using efficient multi-dimensional range matching. ACM SIGCOMM Computer Communication Review **28**(4), 203–214 (1998)
18. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
19. Narayanan, R., Kotha, S., Lin, G., Khan, A., Rizvi, S., Javed, W., Khan, H., Khayam, S.: Macroflows and microflows: Enabling rapid network innovation through a split sdn data plane. In: Software Defined Networking (EWSDN), 2012 European Workshop on, pp. 79–84 (2012)
20. Rifai, M., Huin, N., Caillouet, C., Giroire, F., Lopez-Pacheco, D., Moulierac, J., Urvoy-Keller, G.: Too many sdn rules? compress them with minnie. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–7. IEEE (2015)
21. Rottenstreich, O., Keslassy, I., Hassidim, A., Kaplan, H., Porat, E.: Optimal in/out tcam encodings of ranges. IEEE/ACM Transactions on Networking **24**(1), 555–568 (2016)
22. Rottenstreich, O., et al.: Lossy compression of packet classifiers. In: Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for networking and communications systems, pp. 39–50. IEEE Computer Society (2015)
23. Stephens, B., Cox, A., Felter, W., Dixon, C., Carter, J.: Past: Scalable ethernet for data centers. In: Proceedings of CoNEXT, pp. 49–60. ACM, New York, NY, USA (2012)
24. Suri, S., Sandholm, T., Warkhede, P.: Compressing two-dimensional routing tables. Algorithmica **35**(4), 287–300 (2003)
25. Taylor, D.E.: Survey and taxonomy of packet classification techniques. ACM Computing Surveys (CSUR) **37**(3), 238–275 (2005)
26. Van Zuylen, A.: Linear programming based approximation algorithms for feedback set problems in bipartite tournaments. In: Theory and Applications of Models of Computation, pp. 370–379. Springer (2009)