Public Review for

On the Complexity of Non-Segregated Routing in Reconfigurable Data Center Architectures

K. Foerster, M. Pacut, S. Schmid

To deal with varying traffic demands more effectively, a data center network might include reconfigurable interconnects enabling the network to align its topology closer with the current demand. This paper by Klaus-Tycho Foerster, Maciej Pacut, and Stefan Schmid studies optimal demand-aware routing in a static topology augmented with one switch that can set up reconfigurable links to the other nodes. While the paper extends previous results by the authors, it presents a number of interesting new results on algorithmic complexity of such optimal demand-aware routing. The two main parameters in the model are (1) the maximum simultaneous connections a node has to the reconfigurable switch and (2) the number of alternations, i.e., how many times a path switches between static and reconfigurable links. The rigorous analysis demonstrates that optimal demand-aware routing in the considered hybrid architecture is NP-hard in general. Optimal polynomial-time algorithms exist in the special cases where each path uses at most one reconfigurable link and fully stays within either static topology or reconfigurable portion of the network (i.e., there are no alternations). When a path uses at least two reconfigurable links, optimal polynomial-time solutions without alternations exist only when a node has at most one connection to the reconfigurable switch. With alternations, the problem remains NP-hard regardless of how many connections to the reconfigurable switch are allowed. The analytical results reveal that while reconfigurable interconnects make data center networks more flexible, exploitation of this greater flexibility to route traffic demands more effectively faces steep computational challenges. The analysis also suggests a need for a better understanding on algorithmic complexity of approximate demand-aware routing with provable approximation guarantees.

Public review written by
Sergey Gorinsky
IMDEA Networks Institute, Spain

On the Complexity of Non-Segregated Routing in Reconfigurable Data Center Architectures

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna, Austria
klaus-tycho.foerster@univie.ac.at

Maciej Pacut
University of Wroclaw, Poland
pct@cs.uni.wroc.pl

Stefan Schmid
Faculty of Computer Science
University of Vienna, Austria
stefan_schmid@univie.ac.at

ABSTRACT

By enhancing the traditional static network (*e.g.,* based on electric switches) with a dynamic topology (*e.g.,* based on reconfigurable optical switches), emerging reconfigurable data centers introduce unprecedented flexibilities in how networks can be optimized toward the workload they serve. However, such hybrid data centers are currently limited by a restrictive routing policy enforcing artificial *segregation*: each network flow can only use either the static or the flexible topology, but not a combination of the two.

This paper explores the algorithmic problem of supporting more general routing policies, which are not limited by segregation. While the potential benefits of non-segregated routing have been demonstrated in recent work, the underlying algorithmic complexity is not well-understood. We present a range of novel results on the algorithmic complexity of non-segregated routing. In particular, we show that in certain specific scenarios, optimal data center topologies with non-segregated routing policies can be computed in polynomial-time. In many variants of the problem, however, introducing a more flexible routing comes at a price of complexity: we prove several important variants to be NP-hard.

CCS CONCEPTS

• **Networks → Network architectures**; • **Theory of computation → Design and analysis of algorithms**;

KEYWORDS

Algorithms, Complexity, Routing, Optical Circuit Switches, Free-Space Optics, Reconfigurable Topologies

1 INTRODUCTION

With the increasing popularity of data-centric applications, the design of efficient and cost-effective data center networks has received much attention over the last years. While traditionally, data center topologies are optimized to provide performance guarantees under *arbitrary* workloads (*e.g.,* [2, 22, 23, 31, 36, 48]), emerging *reconfigurable* topologies (*e.g.,* [10, 12, 17, 20, 25, 27, 34, 47, 54]) allow to dynamically adjust the topology, enabling *demand-aware* ("workload-aware"), self-adjusting networks [8]. Demand-aware networks can achieve a performance similar to demand-oblivious networks at lower cost [10, 20], depending on the workload.

However, while reconfigurable topologies introduce a new dimension of flexibility to the data center design problem, it is typically impossible to fully exploit these flexibilities due to restrictive routing policies. Reconfigurable data center networks are often *hybrid* and combine two types of topologies: a static topology which consists of *e.g.,* electric switches, and a flexible topology providing the reconfigurable links, which could be implemented using *e.g.,* optical circuit switches, wireless technology, free space optics or electric solutions as well. But while the topology is hybrid, routing is often not: routing policies enforce an artificial *segregation*. In segregated routing, a network flow can either only use the static topology (*e.g.,* mice flows) or only the flexible topology (*e.g.,* elephant flows), but not a combination of the two; this can lead to a suboptimal resource allocation [19].

This paper is motivated by the desire to unlock the full flexibility of reconfigurable networks by supporting *non-segregated routing*. In particular, we are interested in the algorithmic complexity of supporting such general routing policies, essentially a *joint optimization problem*, involving both topology design and routing.

Contributions. We explore the algorithmic complexity of supporting more general routing policies, which are not limited by segregation. We classify demand-aware routing in reconfigurable networks along two dimensions, (1) the number of connections to the reconfigurable switch per node and (2) if alternations between the static and reconfigurable network parts are allowed. We also investigate the effect of allowing at most one reconfigurable hop per route. A tabular overview of our results is presented in Table 1.

- *Segregated routing:* We first show that when each route is limited to at most one reconfigurable link, an optimal routing can be found efficiently (§3). However, if we remove this restriction, then even allowing $b = 2$ connections to the reconfigurable switch turns the problem NP-hard (§4.1), as well as for every larger $b \in \mathbb{N}$.
- *Non-segregated routing:* When one can mix reconfigurable and static links, routing is more efficient, but computationally harder to optimize. We show that even allowing $k = 1$ alternations between network parts is NP-hard (§4.3), even if the reconfigurable degree is just $b = 1$. We generalize this result to every $b \in \mathbb{N}$ in §4.1 and §4.2.

While these results are presented for the popular model of connecting one reconfigurable switch to the nodes, as we point out in §6, many results transfer to multiple switches. Our results further apply to both uni- and bidirectional reconfigurable links.

2 MODEL

We study the problem of computing a data center topology to optimally serve a given communication pattern, where the topology combines static (fixed) and reconfigurable links. Our notation follows the model of [19] for most parts.

Research partially supported by Polish National Science Centre grant 2016/23/N/ST6/03412.

Reconfig. degree $\Delta_R \in \mathbb{N}$	$h_\# \leq 1$, no alternations	No alternations	With alternations
$\Delta_R = 1$	**P** [19]	**P** [19]	**NPC** ($k = 1$: §4.3, $k > 1$: [19])
$\Delta_R = 2$	**P** (§3)	**NPC** (§4.1)	**NPC** (§4.1)
$\forall \Delta_R > 2$	**P** (§3)	**NPC** (§4.2)	**NPC** (§4.2)

Table 1: Overview of the complexity of demand-aware routing, depending on the reconfigurable degree b (how many connections to the reconfigurable switch) and the number of allowed alternations k between the static and the reconfigurable topology. Most problem classes are NP-hard to optimize, except when adding the restriction that at most $h_\# = 1$ reconfigurable links may be used on a route—a choice that simplifies calculating the routing, but at the cost of routing efficiency.

Network model. Let $N = (V, E, w)$ be a weighted *hybrid* network [35, 50] connecting the nodes $V = \{v_1, \ldots, v_n\}$ (*e.g.,* top-of-the-rack switches), using static links $E = \{e_1, \ldots, e_m\}$ and reconfigurable links implemented through a reconfigurable (optical circuit) switch. A reconfigurable switch connects the set of nodes V by choosing a matching M on V, where two matched nodes are connected by a bidirectional link. For the sake of generality, we assume each link, whether static/fixed or dynamic/reconfigurable, comes with a positive weight w (a cost, *e.g.,* latency).

Traffic demands. The resulting network should serve a certain communication pattern, represented as a $|V| \times |V|$ communication matrix D (the *demand matrix*). An entry (i, j) in D represents the communication frequency from the node v_i to the node v_j.

Optimization objective. We say that the hybrid network N is *configured* by the reconfigurable switch, where the links contained in the matching M are referred to as the *configuration* of N. For ease of notation, we will simply write $N(\mathcal{M})$ to denote the concrete topology resulting from configuration $\mathcal{M}$ and define $dist_{N(\mathcal{M})}(i, j)$ to be the shortest (weighted) distance from node v_i to node v_j on the network $N(\mathcal{M})$. Given a hybrid network N and a communication demand D, our goal is to find a configuration $N(\mathcal{M})$ of the network N that minimizes the (weighted) average path length for serving D in N. Succinctly stated:

$$\min \sum_{(i,j)\in D} D[i, j] \cdot dist_{N(\mathcal{M})}(i, j) \ .$$

That is, we aim to minimize the sum of the weighted (*i.e.,* by flow size and link costs) path lengths: for each ordered pair of nodes $v_i, v_j \in V$, we multiply the (weighted) length of the shortest path $dist_{N(\mathcal{M})}(i, j)$ from v_i to v_j on $N(\mathcal{M})$ with their entry (i, j) in D. We denote this reconfigurable routing problem by RRP.

Problem dimensions. The work in [19] already showed a performance gap between networks with segregated and non-segregated routing, *i.e.,* whether or not the routing may use a combination of static and reconfigurable links. We analyze this distinction from a more fine-grained perspective, namely:

- We introduce a parameter k that defines how often a route may switch between static and reconfigurable links, with $k = 0$ and $k = \infty$ representing the extremes of (non-)segregation.
- We allow nodes to connect more than once to a reconfigurable switch. The number of connections is limited by the hardware available to the node (*e.g.,* the number of optical transmitters and receivers). For a node v, by $\delta_R(v)$ we denote the maximum number of reconfigurable links that v may utilize, and we set $\Delta_R(N) = \max_{u \in V(N)} \delta_R(u)$.
- We also study unidirectional reconfigurable links, where each node v has $\delta_R^{\text{in}}(v)$ incoming and $\delta_R^{\text{out}}(v)$ outgoing reconfigurable links, setting $\delta_R^{\text{in}}(v) + \delta_R^{\text{out}}(v) = \delta_R(v)$.

3 OPTIMALITY FOR SEGREGATED ROUTING

We begin our study with the segregated case (*i.e.,* $k = 0$) and study the parameter Δ_R that defines the maximum simultaneous connections a node has to the reconfigurable switch. We show this variant of RRP to be tractable if we restrict the number of reconfigurable links that may be be used on a route (denoted by $h_\#$) to just one. In other words, the case where one must choose to route each demand between either solely along the static network or along a single reconfigurable link (*e.g.,* for elephant flows).

Our result will make use of weighted u-capacitated b-matching algorithms [32], which compute a maximum weight matching for the case where each node v may match $b_v \leq n$ times, with each link e being allowed to be used at most $u_e \leq u$ times. b-matching algorithms were already proposed for reconfigurable networks, *e.g.,* in [47]: however, there the b-matching is used to assign elephant flows to links, without regards to the static network or providing optimality proofs. Conceptually, our proof is inspired by [19, Theorem 1], where the case of $\Delta_R = b = 1$ was considered.

Theorem 3.1. *Let $\Delta_R \in \mathbb{N}$. The resulting reconfigurable routing problem* RRP *with $k = 0$ alternations and $h_\# = 1$ is in P.*

Proof of Theorem 3.1. We prove the theorem statement by formulating the routing problem as a matching problem. To this end, for each pair of demand entries $d_{i,j}, d_{j,i}$ (possibly of size 0) we compute the non-negative gain $g_{i,j}$ obtained by connecting the nodes i, j in the matching, *i.e.,* the potential route improvement which results from using the reconfigurable link from i to j (recall $h_\# = 1$, $k = 0$), multiplied by the combined size of $d_{i,j}, d_{j,i}$. If a reconfigurable link from i to j may not exist, we set $g_{i,j} = 0$. Then, we construct the complete graph G' with the node set V and link weights $g_{i,j}$, and compute a maximum weighted 1-capacitated (each link may only be used once) Δ_R-matching (with $b_v = \delta_R(v)$) in polynomial time [32], and set the respective matching as the configuration $\mathcal{M}$ of N, ignoring links with $g_{i,j} = 0$. □

Remarks on directed routing. We can extend Theorem 3.1 to apply to unidirectional links as well. To this end, we split each node $v \in V$ into two nodes $v^{\text{in}}, v^{\text{out}}$, where v^{in} takes care of all outgoing demands and reconfigurable links of v, analogously for v^{out}. Matching links that may not exist are assigned a weight of 0, *i.e.,* they provide no benefit.

4 HARDNESS OF NON-SEGREGATION

We continue our study with the non-segregated case. It is known from previous work [19] that RRP is NP-hard for a reconfigurable degree of 1 and multiple alternations.[1] We will now show that for

[1] A careful analysis of [19, §3.2] reveals that RRP is NP-hard for $k = 2$ (or more) alternations with $\delta_R = 1$, even though it is only stated for $k = \infty$ in [19].

any combination of 1) alternations $k \geq 1$ and 2) reconfigurable degree $\delta_R \geq 1$, the routing problem RRP remains NP-hard as well. We start with $\delta_R = 2$ in Section 4.1 and $\delta_R > 2$ in Section 4.2, followed by the more complicated case of $\delta_R = 1$ in Section 4.3.

4.1 Reconfigurable Degree of Two

We start with the scenario where all nodes have a reconfigurable degree of 2 and then extend it to higher degree combinations in Section 4.2. We refer to the reconfigurable routing problem RRP with reconfigurable degree of $\delta_R(v) = 2$ as RRP(2). We show that this variant is NP-complete by reduction from the NP-hard Circular Arrangement [33, §2] problem, abbreviated CA, defined as follows in the notation of [33]: given a weighted graph $G = (\mathcal{N}, A)$, where each (demand) arc from A is non-negative, arrange the nodes $\mathcal{N}$ in a graph cycle with unweighted links, s.t. the weighted path length is minimized. We introduce an auxiliary variant of CA called CA+, where every arc weight is at least 1. In Theorem 4.1 we show that this variant is NP-complete, and in Theorem 4.2 we reduce it to RRP(2). By $A \prec_P B$ we denote the existence of a polynomial-time reduction from problem A to problem B.

Theorem 4.1. CA+ $\prec_P$ CA

Proof. For an instance $I = (\mathcal{N}, A)$ of CA, we create an instance $I^+ = (\mathcal{N}, A^+)$ of CA+ by increasing every arc's weight by 1, *i.e.,* for each pair of nodes (u, v), we set $A^+(u, v) := A(u, v) + 1$. We construct an instance I' of CA+ that simulates I. Formally, for any integer Thr, I has a solution of cost at most Thr iff I^+ has a solution of cost at most $Thr + c(n)$, where $n = |\mathcal{N}|$ and

$$\begin{cases} c(2k) = k^3 \\ c(2k+1) = k^3 + (3/2) \cdot k^2 + k/2 \ . \end{cases}$$

($\Rightarrow$) For any solution S for I of cost at most Thr, we produce the solution S^+ for I^+ by replicating the circular arrangement S. For each pair of nodes (u, v), its (non-weighted) path length $\ell_{(u,v)}$ on a cycle S^+ is identical to the path length for (u, v) on a cycle S. Let $S(u, v)$ and $S^+(u, v)$ be the weighted path length of the arc (u, v) in S and S^+, respectively. The weight increase (over S) of each arc in S^+ is 1, hence $S^+(u, v) = S(u, v) + 1 \cdot \ell_{(u,v)}$. The total cost of S^+ is

$$\sum_{(u,v)} S^+(u, v) = \sum_{(u,v)} (S(u, v) + \ell_{(u,v)}) \leq Thr + \sum_{(u,v)} \ell_{(u,v)} \ .$$

It remains to show that $\sum_{(u,v)} \ell_{(u,v)} = c(n)$, *i.e.,* the total (non-weighted) path length between all pairs of nodes on a cycle with n vertices is $c(n)$. If n is odd, *i.e.,* $2k + 1 = n$, then each node u has two arcs to nodes at distance d for $d = \{1, 2, \ldots, k\}$. The sum of path lengths for arcs that involve u is then $2 \cdot (1 + \ldots + k) = k \cdot (k + 1)$. To obtain the total path length, we sum over all nodes and divide by 2 (we counted each arc twice), and hence for odd n

$$\sum_{(u,v)} \ell_{(u,v)} = n \cdot k \cdot (k+1)/2 = k^3 + (3/2) \cdot k^2 + k/2 = c(n) \ .$$

If n is even, *i.e.,* $2k = n$, then each node u has two arcs to nodes at distance d for $d = \{1, 2, \ldots, k-1\}$, and one arc to the node at distance k. The sum of path lengths for arcs that involve u is then $2 \cdot (1 + \ldots + k - 1) + k = k \cdot (k-1) + k = k^2$. To obtain the total path length, we sum over all nodes and divide by 2 (we counted each arc twice), and hence for even n:

$$\sum_{(u,v)} \ell_{(u,v)} = n \cdot k^2/2 = (2 \cdot k) \cdot k^2/2 = k^3 = c(n) \ .$$

($\Leftarrow$) For any solution S^+ for I^+ of cost at most $Thr + c(n)$, we produce the solution S for I by replicating the circular arrangement S. The proof is equivalent to the ($\Rightarrow$) case. We use $S(u, v) = S^+(u, v) - \ell_{(u,v)}$ to show that S has the cost at most Thr. □

Theorem 4.2. CA+ $\prec_P$ RRP(2)

Proof. For an instance $I = (\mathcal{N}, A)$ of CA+, we produce an instance I' of RRP with an empty static network and demands D equivalent to weights of arcs A. Then, for any integer Thr, I has a solution of cost at most Thr iff I' has a solution of cost at most Thr.
($\Leftarrow$) Consider the solution S' of cost at most Thr for I'. The demand between each pair of nodes is positive, hence the optical link configuration in S' must form a connected graph; otherwise, there would exist a pair of nodes that cannot be routed and the solution would be infeasible. Let C be the reconfigurable link configuration from S'. The degree of C is 2 and it is connected, thus C is a cycle. We construct the solution S for I by setting the arrangement equivalent to the cycle C. The weighted path length of each arc (u, v) is then no more expensive than the cost of routing the demand (u, v), and the solution has the cost at most Thr.
($\Rightarrow$) Consider a solution S for I of cost at most Thr, consisting of a cycle C. We produce the equivalent circular reconfigurable network $N = C$ for I' and route demands by shortest paths. The routing of every demand (u, v) is no more expensive than the weighted cost of an arc (u, v) from S, and hence its total cost is at most Thr. □

Conclusions. By combining Lemma 4.1, Lemma 4.2, and the transitivity of relation $\prec_P$, we obtain that RRP(2) is NP-complete. As Directed Circular Arrangement is also NP-hard [33, §3], the above proof can be directly modified to hold for the unidirectional case with $\delta_R^{\text{in}}(v) = \delta_R^{\text{out}}(v) = 1, \forall v \in V$. Our construction consists of reconfigurable links only, hence the NP-hardness is independent of the number of allowed alternations k.

4.2 Beyond a Reconfigurable Degree of Two

We now introduce techniques that allow us to extend the proofs from Section 4.1 to higher reconfigurable degrees. We believe these techniques also to be of independent interest for future work.

Link enforcement. If we want to force two nodes v, v' to match with each other, we can create an arbitrarily high demand between them, s.t. any optimal solution must match v and v'. With respect to optimal solutions, the link (v, v') must be created, for both the uni- and bidirectional case.

2-Extension technique. Consider a network where every node has the identical reconfigurable degree of exactly two, *i.e.,* $\forall v \in V : \delta_R(v) = \Delta_R = 2$. We will now show, first for the bidirectional case, that if RRP is NP-hard in that specific setting, then it is also NP-hard when the reconfigurable degree is increased to some larger $b \in \mathbb{N}$. Similarly, we can also use this extension technique to extend the reconfigurable degree of some subset of nodes from 2 to b for algorithmic purposes, *i.e.,* that it leaves the matching of an optimal solution untouched and all newly created nodes will have a reconfigurable degree of b. To increase the reconfigurable degree

from 2 to 3, we create a complete binary tree of depth 3 by enforcing links, where we enforce to connect the root to a node v with a connectivity deficit of one, and two links between the leaves of this tree T_v^1 s.t. all 7 nodes $v_1^1, v_2^1, \ldots v_7^1$ in T_v^1 have a reconfigurable degree of 3. For the unidirectional case, we orient the link (v, v_1^1) towards respectively away from v, analogously for the other links, the reconfigurable degree sum δ_R remains unchanged.

We now show how to directly jump from 2 to b: we create $b-2$ trees $T_3, \ldots, T_b$, where we enforce 7 cliques, one for each of the seven node groups $v_1^i, \ldots, v_7^i$—each of them thus having a reconfigurable degree of $3+b-3=b$, except for the v_1^is, which have $2+b-3=b-1$. We then enforce to connect those $b-2$ v_1^is to v. Again, for the unidirectional case, we orient those links arbitrarily.

By applying the 2-extension technique, the earlier theorem can be extended to any fixed reconfigurable degree in $\mathbb{N}$.

Corollary 4.3. *For every number of allowed alternations $k \in \mathbb{N}$ and for every reconfigurable degree $\delta_R(v) = b, b \in \mathbb{N}, b \geq 2, \forall v \in V$ holds: the reconfigurable routing problem* RRP *is NP-complete.*

Furthermore, as the 2-extension technique only increased the number of nodes by a factor of $O(b)$, the reconfigurable degree b can be raised even higher as a function of n, *i.e.,* $b = \lceil f(n) \rceil \geq 2$. As long as this function f remains polynomial, NP-hardness holds.

4.3 Reconfigurable Degree of One

In this section, we show that RRP is NP-complete even in the restricted variant, where all nodes have the reconfigurable degree of 1, and with at most 1 alternation for the routing of any demand. Our construction unfolds in two stages. First, we introduce an auxiliary variant of RRP problem: for any integer ℓ, by ℓ-RRP we denote the variant of RRP, where the reconfigurable network $\mathcal{M}$ consists of at most ℓ links. In Lemma 4.4, we present a polynomial time reduction from RRP to ℓ-RRP. Then, in Lemma 4.5, we reduce the classic Vertex Cover problem to ℓ-RRP.

Lemma 4.4. *For any positive integer ℓ, we have ℓ-RRP $\prec_P$ RRP.*

Proof. Consider any ℓ-RRP instance I with the static network G. We assume that G is normalized, *i.e.,* the minimum weight of a link is 1. We construct an instance I' of RRP that simulates I. Precisely, we prove that for any integer Thr, I has a solution of cost at most Thr iff I' has a solution of cost at most

$$Thr' := Thr + (2 \cdot (\lfloor n/2 \rfloor - \ell)) \cdot ((n-1) \cdot (\mathcal{D}+1) + \mathcal{D}) \cdot (Thr+1) \ ,$$

where $\mathcal{D}$ is the maximum weight of the shortest weighted path between any two nodes in G. We preserve the weight of static links, the weights of reconfigurable links, and the demands between every pair of nodes from G. We introduce an additional set of nodes $\mathbb{A}$ of size $2 \cdot (\lfloor n/2 \rfloor - \ell)$. We connect every node from $\mathbb{A}$ with every node from G by a static link with weight $\mathcal{D}+1$. Every reconfigurable link between $\mathbb{A}$ and G has weight $\mathcal{D}$. We produce additional demands of volume $Thr+1$ from every node from $\mathbb{A}$ to every node from G.

Consider a demand between a pair of nodes $a \in \mathbb{A}$, $b \in V(G)$. The optimal routing of a demand from a to b costs $\mathcal{D}$ if a reconfigurable link (a, b) is present, and costs $\mathcal{D}+1$ otherwise. If the reconfigurable link is not present, every non-direct route costs at least $\mathcal{D}+1$: the cost at least $\mathcal{D}$ is incurred between a and any node $c \in V(G)$, and the cost at least 1 is incurred between b and c (the static network is normalized). Complementary, the optimal route between a and b costs at most $\mathcal{D}+1$, as a direct static link of such weight exists.

Note that providing $\mathbb{A}$ with less than $|\mathbb{A}|$ reconfigurable links results in surpassing the threshold Thr'. As at most one reconfigurable link can be adjacent to any node, each node from $\mathbb{A}$ incurs the cost of at least $((n-1) \cdot (\mathcal{D}+1) + \mathcal{D}) \cdot (Thr+1)$ for its demands. Every node from $\mathbb{A}$ with no adjacent reconfigurable link incurs the cost at least $(n \cdot (\mathcal{D}+1)) \cdot (Thr+1)$, which incurs additional cost at least $Thr+1$ that cannot be compensated by savings in routing demands among nodes in G. As the maximum reconfigurable degree (Δ_R) is 1, in every solution to I' with cost at most Thr', every node from $\mathbb{A}$ has a reconfigurable link to some node in G.

To reconstruct the solution to I, we take the reconfigurable links among nodes from G from the solution to I'. Now, we claim that the reconstructed solution has exactly ℓ reconfigurable links. In any graph with n vertices, the maximum size of any matching is $\lfloor n/2 \rfloor$. To restrict it to ℓ links, we need to remove $\lfloor n/2 \rfloor - \ell$ matching links. To prevent one link from appearing, we need to reduce the number of matchable nodes by 2. Each node from $\mathbb{A}$ matches to one node from G, and $|\mathbb{A}| = 2 \cdot (\lfloor n/2 \rfloor - \ell)$.

As every node of $\mathbb{A}$ has exactly one reconfigurable link to a node from G, the cost of routing demands between $\mathbb{A}$ and G is exactly $|\mathbb{A}| \cdot ((n-1) \cdot (\mathcal{D}+1) + \mathcal{D}) \cdot (Thr+1)$. By the definition of the threshold Thr', the remaining budget for routing demands inside G is Thr. Note that we preserve the shortest paths among nodes from G: by the weight of static and reconfigurable links between $\mathbb{A}$ and G, the routes through $\mathbb{A}$ weigh more than any path in the original network. Hence, the cost of the reconstructed solution to I is at most Thr. □

Lemma 4.5. *It holds that* Vertex Cover $\prec_P \bigcup_\ell \ell$-RRP.

Proof. For an integer t, the decision version of the Vertex Cover is the problem of determining the existence of a vertex cover of size at most t. Consider any decision Vertex Cover instance $\langle G, t \rangle$, where $G = \langle V, E \rangle$. We produce a ℓ-RRP instance (where $\ell = |E| + t$) that has a feasible solution that satisfies a threshold $Thr := 5 \cdot |E|$ iff a vertex cover of G of size at most t exists.

The construction unfolds as follows. For each vertex $v \in V$ we produce a Vertex Gadget that consists of two nodes: a_v and b_v. For each link $e \in E$ we produce a Link Gadget that consists of three nodes: l_e, m_e and r_e, and two links of weight 3: (l_e, m_e) and (r_e, m_e). For each link $e = (u, v) \in E$ we produce two links of weight 2: (m_e, b_u) and (m_e, b_v) and two links of weight 1: (a_u, l_e) and (a_v, r_e). For each link $e \in E$, reconfigurable links (l_e, m_e) and (r_e, m_e) have weight 1 and for each vertex $v \in V$, a reconfigurable link (a_v, b_v) has weight 1. Remaining reconfigurable links $(x, y) \in V \times V$ have weight equal to the shortest path (via static links only) between x and y in graph G, and an appearance of such a reconfigurable link does not improve routing of any demand. For each link $e = (u, v) \in E$ we produce two unitary demands: (m_e, a_u) and (m_e, a_v), and we call those the *cover demands* of e. The construction is depicted in Figure 1.

Consider a demand (m_e, a_u). We distinguish among three ways of routing it: In presence of the reconfigurable link (m_e, l_e), the *short route* of weight 2 consists of nodes $m_e \to l_e \to a_u$. In presence of the reconfigurable link (a_v, b_v), the *medium route* of weight 3 consists of nodes $m_e \to b_u \to a_u$. We classify every other route (of weight ≥ 4) as a *long route*. Symmetrically, for a demand (m_e, a_v), analogous short routes through vertex r_e (instead of l_e) exist.

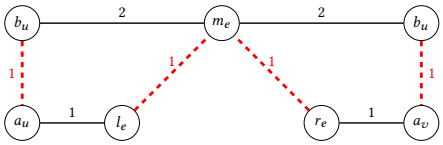

Figure 1: Construction for a link e adjacent to vertices u and v. Static links are drawn solid, and have their weight denoted next to them. Only reconfigurable links that can possibly improve the routing are shown (dashed). We omit the other links and the reconfigurable switch in this figure for clarity.

We say that a vertex $v \in V$ is *active* if the reconfigurable link (a_v, b_v) appears. Now, we argue that at most t vertices are active. Assume that more than t vertices are active. As we have at most $t + |E|$ reconfigurable links, there exists a link $f \in E$ such that none of reconfigurable links $\{(m_f, l_f), (m_f, r_f)\}$ exists. In this case, no short route for the cover demands of f exists, and the cost incurred for them is at least 6. For the remaining cover demands $e \in E \setminus \{f\}$: as $\Delta_R = 1$, at most one reconfigurable link from $\{(m_e, l_e), (m_e, r_e)\}$ exists. Hence, at most one of the cover demands of e can be routed by the short route of cost 2, and the minimum cost of routing of both cover demands of e is 5. Summing up, the total cost is $6 + 5 \cdot (|E| - 1) > Thr$, a contradiction.

To reconstruct the solution to the Vertex Cover, we take active vertices. Now, we argue that such a solution covers all links. Note that any solution that routes any demand by a long route exceeds the threshold. We stated previously that for each $e = (u, v) \in E$, at most one of the cover demands of e is routed by the short route. Hence, exactly one of the cover demands of e is routed by a path of cost 3, and the only path of such weight is the medium route to either u or v. The existence of a medium path implies that either u or v is active, and hence e is covered.

Finally, we show how to reconstruct the ℓ-RRP solution given a vertex cover. Consider a link $e = (u, v) \in E$, and assume that it is covered by u. We route the demand (m_e, a_u) by the medium route, and we route the demand (m_e, a_v) by the short route, placing reconfigurable links to allow the existence of such routes. □

Remarks on directed routing. We can modify Lemmas 4.4 and 4.5 to show hardness in the directed routing model. Instead of setting $\Delta_R = 1$, we set $\delta_R^{\text{in}} = 1$ and $\delta_R^{\text{out}} = 1$ (note that those values are minimal for any reconfigurable links to appear). To show that we can reduce the number of reconfigurable links to ℓ, we modify Lemma 4.4 in the following way: we direct the reconfigurable and static links, and demands between A and G towards nodes of A. As the maximum number of reconfigurable links in the directed routing problem is n (rather than $\lfloor n/2 \rfloor$), we adjust the size of set A to $n - \ell$, and we adjust the threshold value accordingly: $Thr' := Thr + (n - \ell)) \cdot ((n - 1) \cdot (\mathcal{D} + 1) + \mathcal{D}) \cdot (Thr + 1)$. In any solution of the cost at most Thr, each node from A has an incoming link, and the number of links inside G is $n - (n - \ell) = \ell$. Finally, we modify Lemma 4.5 by directing every reconfigurable and static link, and every demand from m_e towards a_v. Note that although the model allows for multiple hops through reconfigurable links, in our construction we used paths with at most one reconfigurable link.

Conclusions. By combining Lemma 4.4, Lemma 4.5, and the transitivity of relation $\prec_P$, we obtain that RRP is NP-complete. The problem remains NP-complete even if we allow at most one alternation, and at most one hop through the reconfigurable network in the routing of any demand.

5 RELATED WORK

Most existing literature on data center network design deals with demand-oblivious topologies, see [41] for a recent survey. In contrast, we in this paper are interested in demand-aware network designs, which not only arise in data centers but also in wide area networks, *e.g.,* [16, 24, 27, 28, 37, 46].

We are not the first to explore non-segregated routing in hybrid networks. In particular, Xia *et al.* [53] leverage converter switches to dynamically convert between a Clos network and approximate random graphs of different sizes. Venkatakrishnan *et al.* [50] show that routing policies restricted to direct or single-hop routing are inefficient and present near-optimal scheduling algorithms, however, only for the segregated case; the general case is stated as an open problem. An orthogonal approach is taken by Mellette *et al.* [38, 39] who consider switches which rotate through a set of pre-defined matchings (building expander-like graphs in [39]), also leveraging Valiant-style [49] multi-hop optical connections.

In this paper, we are particularly interested in network design and routing algorithms which come with *formal (approximation or optimality) guarantees.* Most prior algorithmic works usually assume segregated routing models and rely on heuristics based on matchings [10, 17, 34, 35, 51], edge-coloring [13], or stable-matching algorithms [15, 20], see [29, 52]. Avin *et al.* [5] presented a constant-degree network design algorithm which achieves a constant approximation of the optimal expected route length, which is shown to be proportional to the conditional entropy of the workload. Their approach of combining per-source optimal tree networks, has recently been extended to account also for *congestion* [7]. Furthermore, the authors showed that a connection to coding theory can be leveraged to design *resilient* demand-aware networks. However, the above results (and others [4, 44, 45]) concern fully reconfigurable networks, where *all* links are reconfigurable.

Closer to our work (and reality) are the results by Foerster *et al.* [19] who provide polynomial-time *exact* (*i.e.,* optimal) algorithms, for specific demands and models, and also derive first hardness results. The performance of various heuristics in this setting is evaluated in [18]. We in this paper extend [19] by investigating the complexity of more general non-segregated routing.

The problem of enhancing a given static network with a reconfigurable topology is related to classic combinatorial problems arising in graph theory. For example, Manos *et al.* [42] presented algorithms to augment a given graph with *ghost edges* to provide small world properties and short path lengths, see also the recent paper by Gozzard for a good overview of the state-of-the-art [21]. The underlying problems are also related to the *k-median* problem [40] and known to be hard, even to approximate, in general [43]. Besides considering shortest paths, researchers have also investigated algorithms to reduce the network diameter [11, 14]. In contrast to these works, motivated by emerging optical switches, we consider the problem of adding links via matchings, hence introducing a new perspective on the b-matching literature [1, 30], typically arising in market situations where, *e.g.,* users need to be matched to a cardinality-constrained set of items, *e.g.,* matching children to schools. In this paper, we are only interested in the route length between nodes which actually communicate (demand-aware routing).

Finally, we note that there also exist results on dynamic network design algorithms which aim to strike a balance between reconfiguration costs and providing shorter routes [3, 6, 9, 26, 44, 45], as well as for the case where links need to be removed for maintenance [55].

6 CONCLUSION

This paper showed that more flexible, non-segregated routing policies can introduce additional algorithmic complexities. In particular, we presented algorithms and charted a detailed complexity landscape of non-segregated routing. We hence hope that our results can be useful and provide a more complete picture of the benefits and costs when moving beyond segregated routing.

Even though we focused on the popular model of one reconfigurable switch in this paper [29, 52], the case of multiple such switches is also of importance [38, 53]. Our hardness results naturally transfer to this extension, and in most non-segregated scenarios, there is not much difference between algorithms for one or multiple switches, as multiple reconfigurable switches can be emulated by one switch, combining 1) large weights for not permitted reconfigurable links and 2) fake child nodes for each node to enforce the inter-switch connectivity constraints.

There still remain several interesting open problems for future research. In particular, it will be interesting to shed light on the complexity of *specific* network topologies. Furthermore, while we have focused on exact algorithms, it remains to explore the complexity of (provably) *approximate* algorithms in more depth.

Acknowledgements. We would like to thank the anonymous reviewers of this article for their helpful comments.

REFERENCES

[1] Faez Ahmed et al. 2017. Diverse Weighted Bipartite b-Matching. In *IJCAI*.
[2] Mohammad Al-Fares et al. 2008. A scalable, commodity data center network architecture. In *SIGCOMM*.
[3] Chen Avin et al. 2015. Self-adjusting grid networks to minimize expected path length. *Theor. Comput. Sci.* 584 (2015), 91–102.
[4] Chen Avin, Alexandr Hercules, Andreas Loukas, and Stefan Schmid. 2018. *rDAN*: Toward robust demand-aware network designs. *Inf. Process. Lett.* 133 (2018), 5–9.
[5] Chen Avin, Kaushik Mondal, and Stefan Schmid. 2017. Demand-Aware Network Designs of Bounded Degree. In *DISC*.
[6] Chen Avin, Kaushik Mondal, and Stefan Schmid. 2018. Push-Down Trees: Optimal Self-Adjusting Complete Trees. *CoRR* abs/1807.04613v1 (2018).
[7] Chen Avin, Kaushik Mondal, and Stefan Schmid. 2019. Demand-Aware Network Design with Minimal Congestion and Route Lengths. In *Proc. IEE INFOCOM*.
[8] Chen Avin and Stefan Schmid. 2018. Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks. In *ACM SIGCOMM Computer Communication Review (CCR)*, Vol. 48(5). 31–40.
[9] Chen Avin and Stefan Schmid. 2019. ReNets: Toward Statically Optimal Self-Adjusting Networks. *CoRR* arXiv:1904.03263 (2019).
[10] Navid Hamed Azimi et al. 2014. FireFly: a reconfigurable wireless data center fabric using free-space optics. In *SIGCOMM*.
[11] Davide Bilò, Luciano Gualà, and Guido Proietti. 2012. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theoretical Computer Science* 417 (2012), 12–22.
[12] Kai Chen et al. 2014. OSA: An Optical Switching Architecture for Data Center Networks With Unprecedented Flexibility. *IEEE/ACM Trans. Netw.* 22, 2 (2014), 498–511.
[13] Li Chen et al. 2017. Enabling Wide-Spread Communications on Optical Fabric with MegaSwitch. In *NSDI*.
[14] Erik D. Demaine and Morteza Zadimoghaddam. 2010. Minimizing the diameter of a network using shortcut edges. In *SWAT*.
[15] Nikhil Devanur et al. 2016. *Stable Matching Algorithm for an Agile Reconfigurable Data Center Interconnect.* Technical Report. Microsoft Research.
[16] Ramakrishnan Durairajan et al. 2018. GreyFiber: A System for Providing Flexible Access to Wide-Area Connectivity. *arXiv:1807.05242* (2018).
[17] Nathan Farrington et al. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *SIGCOMM*.
[18] Thomas Fenz et al. 2019. Efficient Non-Segregated Routing for Reconfigurable Demand-Aware Networks. In *IFIP Networking*.
[19] Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. 2018. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *ANCS*.
[20] Monia Ghobadi et al. 2016. ProjecToR: Agile Reconfigurable Data Center Interconnect. In *SIGCOMM*.
[21] Andrew Gozzard, Max Ward, and Amitava Datta. 2018. Converting a network into a small-world network: Fast algorithms for minimizing average path length through link addition. *Information Sciences* 422 (2018), 282–289.
[22] Albert G. Greenberg et al. 2009. VL2: a scalable and flexible data center network. In *SIGCOMM*.
[23] Chuanxiong Guo et al. 2009. BCube: a high performance, server-centric network architecture for modular data centers. In *SIGCOMM*.
[24] Matt Hall, Vijay Chidambaram, and Ramakrishnan Durairajan. 2018. vFiber: Virtualizing Unused Optical Fibers (Extended Abstract). In *NSDI*.
[25] Daniel Halperin et al. 2011. Augmenting data center networks with multi-gigabit wireless links. In *SIGCOMM*.
[26] Sikder Huq and Sukumar Ghosh. 2017. Locally Self-Adjusting Skip Graphs. In *ICDCS*.
[27] Su Jia et al. 2017. Competitive analysis for online scheduling in software-defined optical WAN. In *INFOCOM*.
[28] Xin Jin et al. 2016. Optimizing Bulk Transfers with Software-Defined Optical WAN. In *SIGCOMM*.
[29] Christoforos Kachris and Ioannis Tomkos. 2012. A Survey on Optical Interconnects for Data Centers. *IEEE Commun. Surv. Tutor.* 14, 4 (2012), 1021–1036.
[30] Bala Kalyanasundaram and Kirk Pruhs. 2000. An optimal deterministic algorithm for online b-matching. *Theor. Comput. Sci.* 233, 1-2 (2000), 319–325.
[31] Simon Kassing et al. 2017. Beyond fat-trees without antennae, mirrors, and disco-balls. In *SIGCOMM*.
[32] Adam N. Letchford, Gerhard Reinelt, and Dirk Oliver Theis. 2008. Odd Minimum Cut Sets and b-Matchings Revisited. *SIAM J. Disc. Math.* 22, 4 (2008), 1480–1487.
[33] Vincenzo Liberatore. 2004. Circular arrangements and cyclic broadcast scheduling. *J. Algorithms* 51, 2 (2004), 185–215.
[34] He Liu et al. 2014. Circuit Switching Under the Radar with REACToR. In *NSDI*.
[35] He Liu et al. 2015. Scheduling techniques for hybrid circuit/packet networks. In *CoNEXT*.
[36] Vincent Liu et al. 2013. F10: A Fault-Tolerant Engineered Network. In *NSDI*.
[37] Luo Long et al. 2019. DaRTree: Deadline-Aware Multicast Transfers in Reconfigurable Wide-Area Networks. In *IEEE IWQoS*.
[38] William M. Mellette et al. 2017. RotorNet: A Scalable, Low-complexity, Optical Datacenter Network. In *SIGCOMM*.
[39] William M. Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C. Snoeren, and George Porter. 2019. Expanding across time to deliver bandwidth efficiency and low latency. *CoRR* abs/1903.12307 (2019).
[40] Adam Meyerson and Brian Tagiku. 2009. Minimizing Average Shortest Path Distances via Shortcut Edge Addition. In *APPROX-RANDOM*.
[41] Mohammad Noormohammadpour and Cauligi S Raghavendra. 2017. Datacenter Traffic Control: Understanding Techniques and Tradeoffs. *IEEE Commun. Surv. Tutor.* 20, 2 (2017), 1492–1525.
[42] Manos Papagelis, Francesco Bonchi, and Aristides Gionis. 2011. Suggesting Ghost Edges for a Smaller World. In *CIKM*.
[43] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2015. Selecting shortcuts for a smaller world. In *Proc. SIAM International Conference on Data Mining*. SIAM, 28–36.
[44] Bruna Peres et al. 2019. Distributed Self-Adjusting Tree Networks. In *INFOCOM*.
[45] Stefan Schmid et al. 2016. SplayNet: Towards Locally Self-Adjusting Networks. *IEEE/ACM Trans. Netw.* 24, 3 (2016), 1421–1433.
[46] Rachee Singh et al. 2018. RADWAN: Rate Adaptive Wide Area Network. In *SIGCOMM*. ACM.
[47] Ankit Singla et al. 2010. Proteus: a topology malleable data center network. In *HotNets*.
[48] Ankit Singla et al. 2012. Jellyfish: Networking Data Centers Randomly. In *NSDI*. USENIX Association, 225–238.
[49] Leslie G. Valiant. 1982. A Scheme for Fast Parallel Communication. *SIAM J. Comput.* 11, 2 (1982), 350–361.
[50] Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, and Pramod Viswanath. 2018. Costly circuits, submodular schedules and approximate Carathéodory Theorems. *Queueing Syst.* 88, 3-4 (2018), 311–347.
[51] Guohui Wang et al. 2010. c-Through: part-time optics in data centers. In *SIGCOMM*.
[52] Wenfeng Xia et al. 2017. A Survey on Data Center Networking (DCN): Infrastructure and Operations. *IEEE Commun. Surv. Tutor.* 19, 1 (2017), 640–656.
[53] Yiting Xia et al. 2017. A Tale of Two Topologies: Exploring Convertible Data Center Network Architectures with Flat-tree. In *SIGCOMM*.
[54] Xia Zhou et al. 2012. Mirror mirror on the ceiling: flexible wireless links for data centers. In *SIGCOMM*.
[55] Danyang Zhuo et al. 2017. Understanding and Mitigating Packet Corruption in Data Center Networks. In *SIGCOMM*.