

Article

A Randomized Greedy Heuristic for Steerable Wireless Backhaul Reconfiguration

Nina Skorin-Kapov ^{1,*}, Ricardo Santos ², Hakim Ghazzai ³ and Andreas Kassler ⁴ 

¹ Department of Engineering and Applied Technologies, University Center of Defense, San Javier Air Force Base, MDE-UPCT, 30720 Murcia, Spain

² BISDN GmbH, 10785 Berlin, Germany; ricardo.santos@bisdn.de

³ School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA; hghazzai@stevens.edu

⁴ Department of Mathematics and Computer Science, Karlstad University, 65188 Karlstad, Sweden; andreas.kassler@kau.se

* Correspondence: nina.skorinkapov@tud.upct.es

Abstract: In this paper, we consider the reconfiguration of wireless backhaul networks with mechanically steerable antennas in the presence of changing traffic demands. Reconfiguration requires the scheduling and coordination of several operations, including antenna alignment and link establishment/removal, with minimal disruption to existing user traffic. Previously, we proposed a Mixed Integer Linear Program (MILP) to orchestrate such reconfiguration with minimal packet loss. While the MILP solves the problem optimally for a limited number of discrete reconfiguration time slots, it does not scale well. In this paper, we propose an iterative randomized greedy algorithm to obtain suboptimal solutions in reduced time. The algorithm schedules the reconfiguration of wireless links by ranking them according to a set of attributes with associated weights and selecting them according to a randomized greedy function. Results on six different network scenarios indicate that the proposed algorithm can achieve good quality solutions in significantly less time. Furthermore, by extending the reconfiguration time beyond the maximum number of time slots solvable by the MILP, the proposed heuristic can obtain superior solutions for some problem instances. The number of iterations of the algorithm can be tuned for its applicability in both offline and online planning scenarios.

Keywords: wireless backhaul networks; reconfiguration; randomized greedy heuristic; mechanical steering



Citation: Skorin-Kapov, N.; Santos, R.; Ghazzai, H.; Kassler, A. A Randomized Greedy Heuristic for Steerable Wireless Backhaul Reconfiguration. *Electronics* **2021**, *10*, 434. <https://doi.org/10.3390/electronics10040434>

Academic Editor: Khaled Elleithy

Received: 21 December 2020

Accepted: 5 February 2021

Published: 10 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

To cope with growing mobile traffic predictions, 5G networks and beyond aim to improve usage of the wireless spectrum and leverage dense deployments of small cell (SC) base stations [1]. As it is not feasible and/or cost-effective to connect a massive amount of small cells through optical links to the core network, a promising alternative is the usage of a wireless mesh backhaul that is composed of various millimeter-wave (mmWave) links [2]. By forming multi-hop paths, small cells can forward aggregated user equipment (UE) traffic via mmWave links towards gateway nodes which are connected to the core through fiber cables. This type of architecture has received considerable attention by the 3GPP under the name of Integrated Access and Backhaul (IAB) and has been recognized as a cost-effective alternative to a wired backhaul [3]. With IAB, mmWave relay links are used to connect non-fiber connected nodes (IAB-nodes) to gateway nodes (IAB-donors) through multi-hop paths. IAB has recently been standardized in 3GPP Rel-16 for 5G NR [4].

The flexible nature of wireless links simplifies the reconfiguration of the wireless backhaul topology (in contrast to a fixed wired backhaul) to cope with changing traffic demands or blockage problems. Effective topology reconfiguration could lead to reduced

packet loss, operational costs and/or energy consumption. This has motivated previous work on determining target backhaul topologies to optimize resources, such as power consumption [5,6] or packet delivery time [7]. While such approaches optimize the desired topologies, they do not consider the packet loss incurred to transition between them. Note, reconfiguration should be as seamless as possible to existing UE traffic. In this paper, we consider the problem of orchestrating the reconfiguration of a mmWave backhaul mesh network to change from a current to a target topology with minimal packet loss.

Due to high path loss, mmWave links require highly directional antennas employing effective beamforming techniques to achieve good link quality [8]. Furthermore, flexible beam steering is required to reconfigure (realign) the links. Electronic steering offers several advantages such as fast beam alignment and multiple beamforming, but has higher complexity and a limited angular steering range requiring multiple mmWave interfaces at the same node to achieve wider coverage. For example, the electronically steerable integrated lens antennas proposed in [9] provide $\pm 22^\circ$ and $\pm 35^\circ$ angular sector coverage (depending on lens radius) requiring more than 10 lens antennas to achieve full 360° range. Mechanical steering provides inherent 360° coverage while maintaining gain, but has slower steering capabilities as it requires physical rotation of the antenna [10]. Furthermore, mechanical steerable platforms can be combined with passive lenses or reflect arrays for mmWave applications with reduced hardware complexity [11]. Although electronic steering is envisioned for many emerging mmWave applications [12], for applications with mostly static links, mechanical steering can be a reasonable option due to flexible steering range and resolution and lack of insertion loss [13]. Note, the wireless backhaul considered in this paper is mostly static, where reconfiguration is assumed to occur sporadically due to significant traffic changes, permanent link blockage, or hardware failures, which typically happen in the order of minutes or hours. While both steering techniques may be applied, in this work we consider mechanical steering in which network reconfiguration is more challenging as links require more time to be established.

To orchestrate steerable wireless backhaul reconfiguration, several operations must be scheduled and coordinated, including antenna rotation, link establishment and removal, and associated traffic forwarding. In order to minimize traffic disruption incurred by the reconfiguration process, we assume temporary intermediate links can be established to forward user traffic. In [14], we proposed an exact mixed integer linear program (MILP) for the problem assuming a set of discrete antenna positions, a fixed antenna rotation speed, and a limited reconfiguration time. Although the proposed MILP provides optimal solutions for such a scenario, it does not scale well due to inherent problem complexity. In this paper, we present an iterative randomized greedy heuristic approach to find sub-optimal solutions for larger problem instances in reasonable time, extending out work from [15]. The approach, referred to as Iterative Randomized Greedy Steerable Backhaul Reconfiguration Algorithm (Iter-RG-SBRA), runs several iterations of a randomized greedy procedure to schedule link establishment/removal over a set of discrete time slots, along with the associated antenna rotation movements, based on a set of potential link attributes with varying weights. Traffic forwarding and packet loss in each time slot are calculated using a Linear Program (LP).

We evaluate Iter-RG-SBRA for multiple topologies and compare solution quality and execution time with those obtained by the optimal MILP (for test cases where it could be solved), a sub-optimal MILP (called PVF-MILP) which fixes a subset of the decision variables, and the case of Direct Reconfiguration (DR), where the transition to the target topology is initiated immediately and no intermediate links are formed. Results indicate that Iter-RG-SBRA significantly outperforms PVF-MILP and DR in all cases, and can achieve good sub-optimal solutions in reasonable time. Furthermore, by increasing the maximal allowed reconfiguration time, Iter-RG-SBRA could in some cases find superior solutions to those obtained by the MILP for the maximal reconfiguration time for which the MILP could be solved.

The rest of this paper is structured as follows: Section 2 covers related work, while Section 3 describes the steerable wireless backhaul reconfiguration problem. Section 4 presents the proposed heuristic algorithm along with an illustrative example. Numerical results are detailed in Section 5 and the paper is concluded in Section 6.

2. Related Work

Previous work on establishing and reconfiguring wireless backhaul networks has mostly focused on the calculation and scheduling of desired backhaul topologies to fulfill certain objectives. For example, the work in [7] considers the backhaul design problem as the joint routing and resource allocation one taking mutual interference across simultaneously active backhaul links into consideration to minimize packet delivery time. A multi-stage algorithm for resource allocation and scheduling for both backhaul and access links was developed in [16] and aims to maximize the number of reliable transmission flows. Authors in [17] optimize multi-slot link scheduling given a set of mmWave backhaul links in a 5G network, to minimize the number of time slots where the subset of links scheduled in every time slot can be activated simultaneously. Backhaul scheduling of mmWave links for femto base stations using appropriate relaying mode selection is studied in [18]. The objective is to determine a stopping criteria for relay probing while maximizing the number of concurrent transmissions. In [19], joint resource scheduling and routing for a mmWave picocellular backhaul is solved to maximize the minimum backhaul throughput delivered to all nodes in the network.

For IAB networks, 3GPP has agreed on topology adaptation in order to reconfigure the backhaul to mitigate link blockage or perform load balancing [20]. A Central Unit (CU) can receive feedback on the traffic load and backhaul link quality metrics, gaining awareness of the overall IAB topology and then calculate the optimal configuration. The topology can be adapted dynamically by changing associations between IAB-nodes. A distributed approach to calculate new IAB topologies was presented in [21], where IAB-nodes selected parent nodes based on link signal strength and on the number of hops needed to reach an IAB-donor. In addition, authors in [22] considered existing resource sharing between IAB links when forming backhaul topologies.

Other wireless backhaul reconfiguration work focuses on optimizing energy consumption. For example, authors in [5] calculate new wireless backhaul topologies which allow nodes and interfaces to be powered on/off in order to minimize energy consumption for dynamic traffic demands. In [6], wireless backhaul topologies are created through optimal energy-efficient joint user association, resource allocation, and backhaul routing. The authors in [23] focus on reducing backhaul power consumption by using network traffic predictions to make decisions on scheduling the transmission of links and then updating the traffic routing accordingly.

While all the aforementioned approaches on wireless backhaul scheduling, design or reconfiguration are focused on determining target topologies to optimize resources, they do not optimize the transition between the calculated topologies considering antenna realignment and link establishment time. To the best of our knowledge, our work is the first to consider this problem. In [14], we proposed a MILP formulation for orchestrating wireless backhaul reconfiguration assuming mechanical steering, and demonstrated the associated reconfiguration operations on an experimental testbed using a Software Defined Networking (SDN)-based architecture in [24]. We also proposed an extended MILP model for green backhaul operation by incorporating time constraints on the powering on/off of nodes and link association, as well as a modified model for electronic steering, in [25]. While the MILP models provide optimal solutions, assuming discrete antenna positions, fixed antenna rotation speed and a limited reconfiguration time, they do not scale well and are not applicable for larger scenarios or dynamic network environments. In this paper, we propose an iterative randomized greedy algorithm to orchestrate steerable wireless backhaul reconfiguration for larger problem instances in less time, which is the extension of our initial work in [15].

3. Steerable Backhaul Reconfiguration Problem Statement

3.1. Network Model

In this work, we consider a wireless backhaul network formed by directional mmWave links. The network is composed of a set \mathcal{N} of small cell nodes with given two-dimensional geographical coordinates. A subset $\mathcal{M} \subseteq \mathcal{N}$ of the nodes are gateways, which are connected through fiber links to the core network with assumed unlimited capacity. Each node $n \in \mathcal{N}$ is equipped with a set of network interfaces \mathcal{I} . Each interface is comprised of a mechanically steerable antenna with 360° azimuth coverage and a mmWave transceiver to establish a mmWave link. For simplicity, we assume that antennas have a discrete set of azimuth angular positions with granularity θ and rotate with a fixed rotation speed. To form a mmWave wireless backhaul link between two nodes, they must be within line-of-sight of each other and their interfaces (antennas) must be properly aligned. We focus only on horizontal alignment in this paper, assuming nodes are in close vicinity with each other and, thus, have the same altitude. Whether or not antennas are capable of aligning with each other can be pre-calculated based on distance and obstacles between them, determined actively by transmitting/receiving probe frames while rotating, or determined passively by using out-of-band 2.4/5 GHz radios to infer the direction of existing neighbors [26]. In our model, we consider link feasibility between nodes n and $n' \in \mathcal{N}$ to be pre-calculated based on nodal position and given as an input parameter in the form of binary matrix $\mathbf{L} = (L_{nn'})$. Antennas do not transmit during rotation to avoid interference issues and link establishment is assumed to be immediate once interfaces are aligned. The associated required alignment angles are given in matrix $\mathbf{V} = (V_{nn'})$ in multiples of θ .

Each backhaul mesh node $n \in \mathcal{N}$ serves traffic demand ρ_n (measured in Mbps), which is given by the aggregated demands of all the UEs served by that node. In our model, we only consider downlink traffic since we assume that upstream traffic does not have high bandwidth requirements and can share the same downlink paths. Consequently, the backhaul paths are formed by unidirectional links from the gateway node(s) in \mathcal{M} (where traffic is injected from the core) to the rest of the topology. Each backhaul link connecting nodes n and n' is characterized by a maximum achievable data rate $R_{nn'}$, given by the average statistics of the channel, based on the path loss due to propagation and atmospheric conditions. We ignore the effects of fast-fading, as we focus on a longer time period of network operation, in comparison to channel coherence time.

3.2. Problem Definition

Our wireless backhaul reconfiguration problem is summarized in Table 1 and described below. Given is a wireless backhaul network in accordance with the model described in Section 3.1, along with its current (initial) topology \mathbf{X}^1 in terms of the established wireless links and the initial angular positions of all interfaces at all nodes \mathbf{A}^1 in multiples of θ . We assume that the set of traffic demands (\mathbf{a}) is such that the current topology cannot serve all the demands resulting in packet loss and motivating the need for reconfiguration. A target topology \mathbf{X}^T is also given as an input parameter, which can serve all the demands with zero loss.

The objective of the problem is to orchestrate the transition from \mathbf{X}^1 to \mathbf{X}^T with minimal packet loss by scheduling antenna rotation and link establishment/removal over time, along with the associated traffic forwarding, during the reconfiguration process. We assume intermediate temporary links (which are not part of the current or target topology) can be established during the reconfiguration in order to perform the transition with minimal disruption to active UE traffic. The link schedule must take into consideration the necessary antenna (interface) rotation movements and link feasibility. Note, rotating all relevant interfaces immediately towards the target topology gives a trivial solution to the problem but would cause several initial links to be torn down simultaneously causing high packet loss during the reconfiguration. By allowing a slower transition with the gradual tearing down of links and establishing temporary links using free interfaces, the total reconfiguration packet loss can be reduced. An example of an initial and target

configuration for a 19 node network with a hexagonal SC deployment, 1 gateway node and 3 interfaces per node is shown in Figure 1. Note, positions of unused interfaces are arbitrary.

Table 1. Steerable backhaul reconfiguration problem definition and notation.

Input Parameters and Notation	
$n, n' \in \mathcal{N}$	Set of SC backhaul nodes
$m \in \mathcal{M} \subseteq \mathcal{N}$	Set of gateway nodes
$i, i' \in \mathcal{I}$	Set of interfaces available at each backhaul node
$t \in \mathcal{T} = \{1, \dots, T\}$	Set of time slots available for reconfiguration
τ	Duration of each time slot (in s)
θ	Granularity of interface antenna angular positions (in degrees)
$\mathbf{a} = (\rho_n) \in \mathbb{R}^+$	Traffic demand of node n (in Mbps)
$\mathbf{L} = (L_{nn'}) \in \{0, 1\}$	Binary matrix indicating whether nodes n and n' can potentially form a link
$\mathbf{R} = (R_{nn'}) \in \mathbb{R}^+$	Maximum achievable data rate (in Mbps) on a potential link between nodes n and n'
$\mathbf{V} = (V_{nn'}) \in \{0, 1, \dots, \frac{360}{\theta}\}$	Interface angular positions (in multiples of θ) required at node n to form a link with node n'
$\mathbf{A}^1 = (A_{ni}^1) \in \{0, 1, \dots, \frac{360}{\theta}\}$	Initial angular positions (in multiples of θ) of all interfaces at all nodes in time slot 1
$\mathbf{X}^1 = (X_{nn'i'}^1) \in \{0, 1\}$	Links active in time slot 1 (the initial topology)
$\mathbf{X}^T = (X_{nn'i'}^T) \in \{0, 1\}$	Links active in time slot T (the target topology)
Solution Output Variables	
$\mathbf{X} = (X_{nn'i'}^t) \in \{0, 1\}$	Links active in each intermediate time slot $t, t = \{2, \dots, T - 1\}$
$\mathbf{C}^+ = (C_{ni}^+) \in \{0, 1\}$	Whether interface i at node n is rotating clockwise in time slot $t, t = \{1, \dots, T - 1\}$
$\mathbf{C}^- = (C_{ni}^-) \in \{0, 1\}$	Whether interface i at node n is rotating counter clockwise in time slot $t, t = \{1, \dots, T - 1\}$
$\mathbf{g} = (g_m^t) \in \mathbb{R}^+$	Data rate (in Mbps) at which each gateway node $m \in \mathcal{M}$ injects traffic into the backhaul mesh network in each time slot $t \in \mathcal{T}$
$\mathbf{r} = (r_{nn'i'}^t) \in \mathbb{R}^+$	Data rate (in Mbps) on link between interface i of node n to interface i' of node n' in each time slot $t \in \mathcal{T}$
$\mathbf{l} = (l_n^t) \in \mathbb{R}^+$	Packet loss rate (in Mbps) at each node n in each time slot $t \in \mathcal{T}$
Objective	
$\min \mathcal{F} = \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \tau \cdot l_n^t$	Total reconfiguration packet loss (in GB)

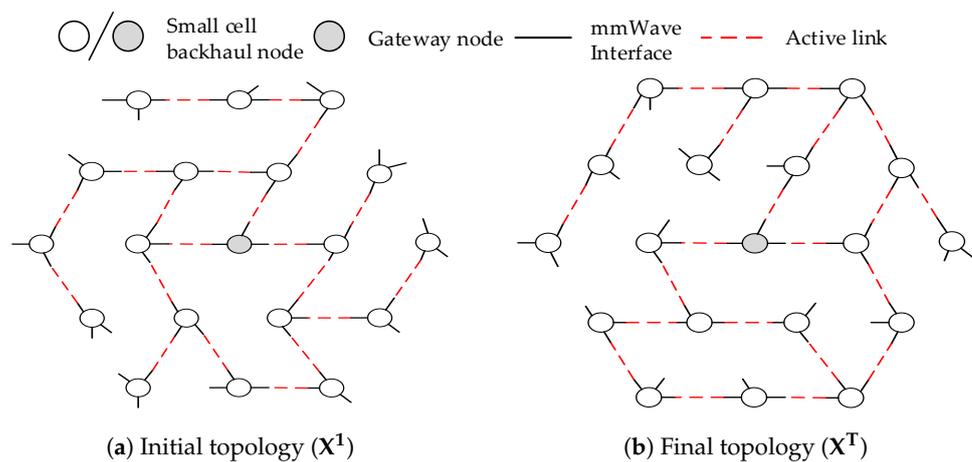


Figure 1. An example of initial and target backhaul topologies for a 19-node hexagonal small cell (SC) deployment with one gateway.

3.3. Time Discretization

To model the problem, we assume the reconfiguration time is limited and divided into $t = 1, \dots, T$ discrete time slots. Note, the reconfiguration time limit (number of time slots) is incorporated into the problem to reduce problem complexity but is not necessarily a strict constraint to the problem. For a given reconfiguration scenario, a longer reconfiguration time can allow for a smoother transition and less overall packet loss as links can be torn down gradually. The total reconfiguration time is assumed to be at least large enough to

accommodate all necessary reconfiguration steps associated with direct reconfiguration, that is, the sum of the time required to rotate all antenna interfaces from their initial positions to the target topology positions if they all start rotating immediately.

The initial topology is active in time slot 1 and the target topology should be active at the latest in time slot T . The length of each time slot, denoted by τ , is determined by the speed of antenna rotation and the granularity of the interface antenna angular positions θ , such that an interface can rotate exactly θ degrees (i.e., change to a neighboring angular position) in a single time slot. Thus, a solution is given in terms of active links (\mathbf{X}), the associated antenna rotation movements ($\mathbf{C}^+, \mathbf{C}^-$), and the traffic forwarding (\mathbf{g}, \mathbf{r}) in each time slot as described in Table 1. The sum of the packet loss over all nodes and all time slots gives the total reconfiguration packet loss \mathcal{F} .

3.4. SDN-Based Orchestration of Steerable Backhaul Reconfiguration

To coordinate such complex orchestration of reconfiguration operations over all nodes and links, a centralized SDN-based approach could be employed. SDN separates the control plane from the data plane, shifting the configuration decisions to a logically centralized controller with a global view of the network. The adoption of heterogeneous network (HetNet) architectures, where legacy 4G LTE macro-cells co-exist with new 5G SCs, is considered an important solution to meet increasing network capacity demands and provide high coverage in 5G networks [27]. This leverages the separation of the control and data planes, where the SCs could provide a high capacity data plane, while the control plane could be offloaded to the macro cells [28]. The system architecture of such an SDN-based mmWave SC backhaul network is shown in Figure 2. All backhaul nodes are assumed to operate under the umbrella of at least one macro cell providing an out-of-band control channel [29]. The SDN controller could solve the optimization problem by outsourcing the complex computations to a dedicated computational entity, denoted as the Backhaul Orchestrator (BO) in Figure 2, which could even run on an external node. The output of the BO could then be used by the SDN controller to re-configure the backhaul nodes and interfaces appropriately (including requesting antenna rotations and link configurations) and push new flow rules to reroute traffic. In [24], we experimentally demonstrated such SDN-based mmWave wireless reconfiguration on a testbed with 4 SC nodes, each equipped with two steerable mmWave interfaces.

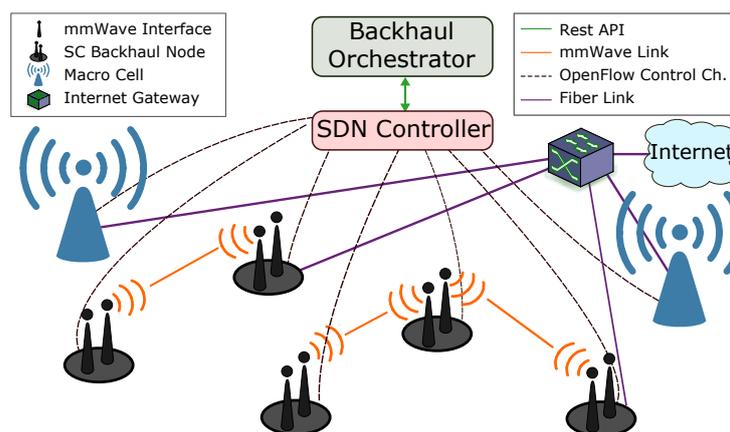


Figure 2. The system architecture of an Software Defined Networking (SDN)-based mmWave backhaul heterogeneous network.

4. A Randomized Greedy Heuristic for Steerable Backhaul Reconfiguration

In [14], we proposed an exact MILP for the described steerable wireless backhaul reconfiguration problem. Since the problem is NP-hard, herein we propose a heuristic which can solve larger instances in reduced time. The heuristic, denoted as Iter-RG-SBRA (Iterative Randomized Greedy—Steerable Backhaul Reconfiguration Algorithm),

iteratively applies a randomized greedy approach with varying weights to determine a possible evolution of wireless links and their associated antenna rotation movements in each time slot. A linear program (LP) is used to determine the traffic forwarding and packet loss for the resulting topologies. In Section 4.1, we first describe the randomized greedy approach that is run in each iteration, denoted as RG-SBRA (Randomized Greedy—Steerable Backhaul Reconfiguration Algorithm). We then detail the structure of the full Iter-RG-SBRA algorithm in Section 4.2.

4.1. RG-SBRA

The main goal of the randomized greedy algorithm RG-SBRA is to schedule and coordinate the tearing down and setting up of links during the reconfiguration process in order to transition from the initial topology (X^1) to the final topology (X^T) by deciding: (i) how long to maintain each initial link, (ii) how soon to establish each final link, and (iii) whether and when to establish intermediate temporary links. This involves coordinating the rotation of all associated interfaces to ensure feasibility of the link and antenna rotation schedule. RG-SBRA sequentially selects links to maintain/establish during the reconfiguration from a sorted list of candidate links according to a randomized greedy function. Candidate links are ranked according to different attributes and associated weights. The set of attributes defined for all possible links ($n_i n'_j$) between interfaces is outlined in Table 2 and described below.

Table 2. Possible link attributes.

Attribute	Description
f_1	Earliest time slot in which the link could be established
f_2	Maximum number of time slots the link could be active
f_3	Whether the link is active in the initial topology
f_4	Whether the link is active in the target topology
f_5	Link utilization in the initial topology
f_6	Link utilization in the final topology
f_7	Number of initially unused interfaces

4.1.1. Link Attributes

$f_1(n_i n'_j)$:

This attribute represents the earliest time slot in which link ($n_i n'_j$) could be established. If the link is part of the initial topology ($X_{n_i n'_j}^1 = 1$) it can be active from the first time slot. Otherwise, $f_1(n_i n'_j)$ is calculated from the minimal angular distance (whether clockwise or counterclockwise) between the initial position of each interface ($A_{n_i}^1, A_{n'_j}^1$) to their required positions in the considered link ($V_{n_i n'_j}, V_{n'_j n_i}$). These values are calculated in multiples of θ to give the number of time slots required to rotate each interface accordingly. Since the link can only be established after both interfaces are aligned, the larger of the two rotation times determines the attribute value. If one or both interfaces of the potential link are part of intermediate links which have already been selected by the algorithm, then this attribute must consider the time the interfaces will be “occupied” by the intermediate link(s) and the number of time slots required to rotate from their position in the intermediate link to their final position. Since it is desirable to select links with faster set-up times, the attribute values are calculated as negative, and then normalized to range [0,1] such that links which take the longest to be established have a value of 0 and links which can be set up immediately have a value of 1.

$f_2(n_i n'_j)$:

This attribute calculates the maximum number of time slots link ($n_i n'_j$) could be active, if it is established. It is desirable to establish links which can be active for a longer period of time, particularly if they are transitional intermediate links for temporarily forwarding user traffic when existing links are torn down and are rotating towards their final positions.

If the link is part of the final topology, or neither interface is part of a final topology link, then $f_2(n_i n'_i)$ is simply calculated as the total number of time slots (T) subtracted by the link establishment time (from $f_1(n_i n'_i)$). However, if one or both of the interfaces belong to other links in the final topology, the link can only be active until these interfaces have to start rotating towards their final positions. Thus, $f_2(n_i n'_i)$ is calculated as the total number of time slots (T) subtracted by the link establishment time (from $f_1(n_i n'_i)$) and the time required to rotate the interface(s) from their angular position(s) in $(n_i n'_i)$ to their final position(s) in the target topology. The values are then normalized to range $[0,1]$.

$f_3(n_i n'_i)$:

This attribute is equal to $X_{nin'i'}^1$ indicating whether the link is part of the initial topology. Maintaining some initial links active for longer is desirable to avoid abrupt packet loss as those links can still serve traffic while other interfaces are rotating to establish intermediate or final topology links.

$f_4(n_i n'_i)$:

This attribute is equal to $X_{nin'i'}^T$ indicating whether the link is part of the final topology. Establishing final links as soon as possible is desirable since these links ultimately lead to zero packet loss.

$f_5(n_i n'_i)$:

This attribute reflects link utilization in the initial topology and is meant to avoid simultaneously tearing down heavily loaded initial links. To calculate this attribute, the Packet Loss LP (see Section 4.1.3) is solved for the initial topology (\mathbf{X}^1) and only a single time slot to determine the data rates on each link in time slot 1 ($r_{nin'i'}^1$). If the link is part of the initial topology ($X_{nin'i'}^1 = 1$), $f_5(n_i n'_i)$ is calculated as the data rate on the link in the first time slot ($r_{nin'i'}^1$) divided by the maximum achievable data rate ($R_{nn'}$). It is considered desirable to maintain the most heavily loaded links from the initial topology as long as possible to avoid causing abrupt high packet loss during the reconfiguration. Conversely, if one or both interfaces of link $(n_i n'_i)$ are associated with other links in the initial topology, the attribute is calculated the negative sum of the link utilization on the respective links. The motivation is that establishing a link which would tear down one or two heavily loaded links is not considered desirable. All values are then normalized to range $[0,1]$.

$f_6(n_i n'_i)$:

This attribute reflects the link utilization in the final topology. The motivation for this attribute is that it is considered desirable to establish the most heavily loaded links from the final topology as soon as possible. To calculate this attribute, the Packet Loss LP (see Section 4.1.3) is solved for the final topology (\mathbf{X}^T) and a single time slot to determine the data rates on each link ($r_{nin'i'}^T$) in time slot T . If the link is part of the final topology ($X_{nin'i'}^T = 1$), then $f_6(n_i n'_i)$ is calculated as the data rate on the link in time slot T ($r_{nin'i'}^T$) divided by the maximum achievable data rate ($R_{nn'}$) giving a value in $[0,1]$. Links which are not part of the final topology have a value of 0 for this attribute.

$f_7(n_i n'_i)$:

This attribute describes the number of interfaces of link $(n_i n'_i)$ which are free in the initial topology, where $f_7(n_i n'_i)$ is 0 if neither interface is free, 0.5 if one interface is free, and 1 if both interfaces are free. If both interfaces are free in the initial topology, establishing a link between the involved nodes does not imply tearing down other links and is, thus, more desirable. Note, if the link itself is part of the initial topology, this attribute still takes on a value of 0 even though no links are torn down. Favoring links which are part of the initial topology is already reflected in attribute f_3 (and indirectly with attributes f_1 and f_5).

4.1.2. RG-SBRA Pseudocode

The pseudocode of RG-SBRA pseudo-code is given in Algorithm 1 and detailed below. RG-SBRA first sets the values of all output variables to 0 (line 2) and creates

a list of all possible links between interfaces based on LOS feasibility ($L_{nn'}$) and time constraints (lines 4–5). The list includes all links from the initial and final topologies and all feasible potential temporary links. Temporary link $(n_i n'_i)$ is only included if it is feasible ($L_{nn'} = 1$) and has sufficient time to be established and maintained for at least 1 time slot ($f_2(n_i n'_i) > 0$). Also, if a link exists in the initial and/or final topology between nodes n and n' , but uses a different pair of interfaces, $(n_i n'_i)$ is not included in the list. Note, links are considered symmetric, that is, $(n_i n'_i)$ and $(n'_i n_i)$ represent the same link. Each link is assigned a score based on its attributes (f_1, \dots, f_7) and associated weights (w_1, \dots, w_7) (line 6).

Algorithm 1: The Pseudo Code of RG-SBRA

Input:Problem input parameters: $\mathcal{N}, \mathcal{M}, \mathcal{I}, \mathcal{T}, \tau, \theta, \mathbf{a}, \mathbf{L}, \mathbf{R}, \mathbf{V}, \mathbf{X}^1, \mathbf{X}^T, \mathbf{A}^1$;Algorithm input parameters: $\mathbf{w} = (w_j), \alpha$, seed**Output:** $\mathbf{X}, \mathbf{C}^+, \mathbf{C}^-, \mathbf{g}, \mathbf{r}, \mathcal{F}$

/* Initialization and Pre-processing phase */

 $PossibleLinks \leftarrow \emptyset$;Set all values of $\mathbf{X}, \mathbf{C}^+, \mathbf{C}^-$ to 0;**foreach** $(n_i n'_i), n, n' \in \mathcal{N}, i, i' \in \mathcal{I}$ **do**
if $L_{nn'} == 1, f_2(n_i n'_i) > 0$, and \mathbf{X}^1 and \mathbf{X}^T do not already include a link between n and n' using a different pair of interfaces **then**
 $PossibleLinks \leftarrow (n_i n'_i)$;

 $Score(n_i n'_i) \leftarrow \sum_{j=1}^7 w_j * f_j(n_i n'_i)$;
end**end**

/* Randomized greedy phase */

while $PossibleLinks \neq \emptyset$ **do**Choose link $(n_i n'_i)$ randomly from α top scored links in $PossibleLinks$;Update $X_{nn'}^t$ for $t = 2, \dots, T - 1$, and associated $C_{tni}^+, C_{tn'i'}^+, C_{tni}^-$ and $C_{tn'i'}^-$ values $t = 1, \dots, T - 1$ such that link $(n_i n'_i)$ is established as soon as possible and active for as long as possible;Remove link $(n_i n'_i)$ and all initial and intermediate links containing interfaces n_i or n'_i from $PossibleLinks$;Update attributes f_1 and f_2 for final links containing interfaces n_i or n'_i from $PossibleLinks$ and recalculate their $Score$;**end**

/* Packet loss calculation phase */

Calculate $\mathbf{g}, \mathbf{r}, \mathcal{F}$ for topologies \mathbf{X} by solving Packet Loss LP;**return** $\mathbf{X}, \mathbf{C}^+, \mathbf{C}^-, \mathbf{g}, \mathbf{r}, \mathcal{F}$;

While the list of possible links is not empty, the algorithm selects a link $(n_i n'_i)$ randomly among the α top scored links (line 10). Note, parameter α can be set to 1 for a pure greedy approach, or >1 for a randomized greedy approach. Link $(n_i n'_i)$ is then established as soon as possible and maintained active for as long as possible which defines the values for $X_{nn'}^t, t = 2, \dots, T - 1$ as well as associated rotation values $C_{tni}^+, C_{tn'i'}^+$ and $C_{tni}^-, C_{tn'i'}^-$ for $t = 1, \dots, T - 1$. (line 11). The selected link can be either (i) an initial topology link, (ii) a final topology link, or (iii) an intermediate link. If the link is an initial topology link, it is active in time slot 1 and remains active either for the entire reconfiguration period, if its interfaces are not involved in any links in the target topology, or until the latest time slot before at least one of its interfaces must start rotating towards its final position. If the link is a final topology link, the link is established as soon as both interfaces can be aligned and it remains active for the rest of the reconfiguration time period. If the link is an intermediate link, the link is established as soon as both interfaces can be aligned and remains active until at least one of its interfaces has to start rotating towards its final position.

Link $(n_i n'_{i'})$ is then removed from the set of possible links, along with the initial and intermediate links which contain interfaces n_i or $n'_{i'}$ as they can no longer be selected (line 12). Attributes f_1 and f_2 of final links containing interfaces n_i or $n'_{i'}$ are updated and their scores are recalculated (line 13). The process is then repeated for as long as there are possible links available. Once the evolution of wireless topologies (\mathbf{X}) is determined as outlined above (all links are defined over all time slots), the associated data rates and packet loss in each time slot is calculated optimally using the Packet Loss LP described in the next subsection (line 15). The final solution of RG-SBRA is returned in terms of the set of active wireless links, the rotation movements of all interfaces over time, data rates and total packet loss.

4.1.3. Packet Loss LP

To determine the data rates and packet loss for a given evolution of wireless links $(\mathbf{X} = (X_{nin'i'}^t))$ and a set of traffic demands (\mathbf{a}) , we solve an LP adapted from the flow conservation constraints of the MILP formulations given in [14,25]. The input parameters are $\mathcal{N}, \mathcal{M}, \mathcal{I}, \mathcal{T}, \tau, \mathbf{a}, \mathbf{R}$, as well as the topologies $\mathbf{X} = (X_{nin'i'}^t)$ for all time slots $t \in \mathcal{T}$, which were found in the randomized greedy phase of RG-SBRA. The decision variables are $\mathbf{g}, \mathbf{r}, \mathbf{l}$ (See Table 1 for full description).

$$\min \mathcal{F} = \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \tau \cdot l_n^t \tag{1}$$

$$0 \leq g_n^t, \forall t, n \in \mathcal{M} \tag{2}$$

$$\sum_{n \in \mathcal{M}} g_n^t \leq \sum_{n' \in \mathcal{N}} \rho_{n'}, \forall t \tag{3}$$

$$0 \leq r_{nin'i'}^t \leq R_{nn'}, \forall n, i, n', i', t \tag{4}$$

$$\begin{aligned} \sum_{n' \in \mathcal{N}} \sum_{i' \in \mathcal{I}} \sum_{i \in \mathcal{I}} X_{n'i'ni}^t r_{n'i'ni}^t + l_n^t = \\ \sum_{n' \in \mathcal{N}} \sum_{i' \in \mathcal{I}} \sum_{i \in \mathcal{I}} X_{nin'i'}^t r_{nin'i'}^t + \rho_n, \forall n \in \mathcal{N} \setminus \mathcal{M}, \forall t \end{aligned} \tag{5}$$

$$\begin{aligned} g_n^t + \sum_{n' \in \mathcal{N}} \sum_{i' \in \mathcal{I}} \sum_{i \in \mathcal{I}} X_{n'i'ni}^t r_{n'i'ni}^t + l_n^t = \\ \sum_{n' \in \mathcal{N}} \sum_{i' \in \mathcal{I}} \sum_{i \in \mathcal{I}} X_{nin'i'}^t r_{nin'i'}^t + \rho_n, \forall n \in \mathcal{M}, \forall t \end{aligned} \tag{6}$$

$$0 \leq l_n^t \leq \rho_n, \forall n, t. \tag{7}$$

The constraint in (2) ensures that ingress rates are non-negative, and that in (3) ensures that the total ingress rate can not exceed the total demand of all the nodes in \mathcal{N} . Constraint (4) ensures that the link rate is non-negative and does not exceed the maximum achievable data rate. Constraints (5) and (6) are the flow conservation constraints at non-gateway and gateway nodes, respectively. At non-gateway nodes, incoming traffic arrives on active links in each time slot, part of which serves the traffic demand of the node while the rest transits the node as outgoing traffic on active links. The difference between the incoming and outgoing traffic rates on the links is that which serves the traffic demand of the node (ρ_n). If this difference is equal to the traffic demand, then the demand can be served in entirety. However, if this difference is less than the traffic demand at the node, only part of the demand can be met while the rest represents the loss rate at the node (l_n^t). This is described by flow conservation Constraint (5) which indicates that for non-gateway nodes, the total input traffic rate and packet loss rate must be equal to the total output rate and traffic demand of the node. For gateway nodes $n \in \mathcal{M}$, the incoming traffic rate

also includes the rate at which traffic is injected from the core into the into the backhaul network at the node (g_n^t) as given in constraint (6). Finally, constraint (7) ensures that the loss rate of each node during each time slot is non-negative and does not exceed its total traffic demand.

4.1.4. Example

We illustrate the calculation of link attributes and the possible link list in RG-SBRA on a small example given in Figure 3. Note, in this example there are 2 gateway nodes and only 2 non-gateway nodes for simplicity. In a realistic scenario, the ratio of non-gateway to gateway nodes would be significantly higher and multihop backhaul paths would be formed. We assume a large granularity of $\theta = 45^\circ$ for easier calculation and illustration of interface positions (the initial angular positions of all interfaces are shown) and assume $T = 6$ reconfiguration time slots. Furthermore, all node pairs can establish a link, except node pair (2 – 4), that is, $L_{24} = L_{42} = 0$. The date rates hypothetically solved by the Packet Loss LP (r_{min}^t) and maximal achievable data rates (R_{nr}) on each link in the initial and final topology are also shown in parentheses.

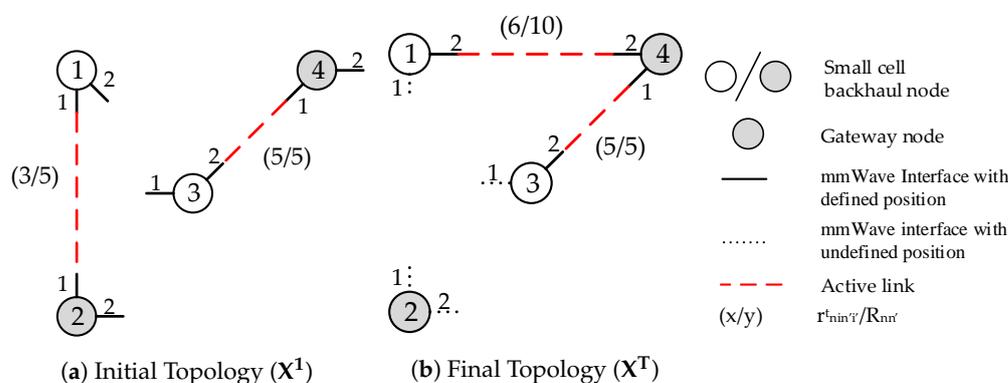


Figure 3. Example initial and final backhaul topologies.

Table 3 gives the attributes for all possible links and shows the link score assuming all weights are equal $w_i = 1, i = 1, \dots, 7$. For example, let us consider link $(1_2 3_2)$. To establish the link, interface 1_2 is already in position, but interface 3_2 would have to rotate 2 positions counterclockwise, and could thus be established at the earliest in time slot $t = 3$. After calculating all the values for all links as negative and normalizing them to interval $[0,1]$, the value for $f_1(1_2 3_2) = 0.5$. To see how long the link could be active, f_2 , we must consider that interface 3_2 is part of a final topology link $(3_2 4_1)$, and would need 2 time slots to rotate from its position in link $(1_2 3_2)$ to its final position in link $(3_2 4_1)$. Thus, $f_1(1_2 3_2) = T - f_1(1_2 3_2) - 2 = 2$, giving a value of 0 after normalization (i.e., it is one of the links which can be active for the shortest number of time slots). Attributes $f_3(1_2 3_2)$ and $f_4(1_2 3_2)$ are equal to zero as the link is neither an initial nor final topology link. Attribute $f_5(1_2 3_2)$ is calculated as the negative link utilization of link $(3_2 4_1)$ which is $-(5/5) = -1$ (0.23 after normalization). Attribute $f_6(1_2 3_2) = 0$, as it is not a final topology link and $f_7(1_2 3_2) = 0.5$ since only interface 1_2 is free in the initial topology.

Figure 4 illustrates the evolution of the possible link list and link selection in the RG-SBRA algorithm for different values of α . In the pure greedy approach ($\alpha = 1$) shown in Figure 4a, the highest scored link is chosen in each step. In the first step, initial link $(3_2 4_1)$ is chosen, and it is established as active in all time slots. Thus, other links using interfaces 3_2 or 4_1 are removed from the list. In the second step, initial link $(1_1 2_1)$ is chosen and can also be established for all time slots as its interfaces are not used in the final topology. In the third step, intermediate link $(2_2 3_1)$ is selected and established in time slot $t = 2$ (it requires 1 time slot to align) and can be active until the end of the reconfiguration period. Lastly, final link $(1_2 4_2)$ is chosen and starts rotating immediately to its position. As it needs 4 time slots to rotate interface 4_2 into position, it can be active starting from time slot $t = 5$. We can see that the algorithm creates a smooth transition by maintaining initial links and

establishing an intermediate link early on in the reconfiguration period to serve traffic while final link (1_24_2) is aligning.

Table 3. List of possible links and their associated attributes and scores.

Possible Link	Link Attributes							Score ($w_i = 1$)
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
(1_12_1)	1.00	1.00	1.00	0.00	0.85	0.00	0.00	3.85
(1_13_1)	0.75	0.75	0.00	0.00	0.38	0.00	0.50	2.28
(1_13_2)	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50
(1_23_1)	0.75	0.50	0.00	0.00	0.62	0.00	1.00	2.87
(1_23_2)	0.50	0.00	0.00	0.00	0.23	0.00	0.50	1.23
(1_24_2)	0.00	0.00	0.00	1.00	0.62	0.60	1.00	3.22
(2_23_1)	0.75	1.00	0.00	0.00	0.62	0.00	1.00	3.37
(3_24_1)	1.00	1.00	1.00	1.00	1.00	1.00	0.00	6.00

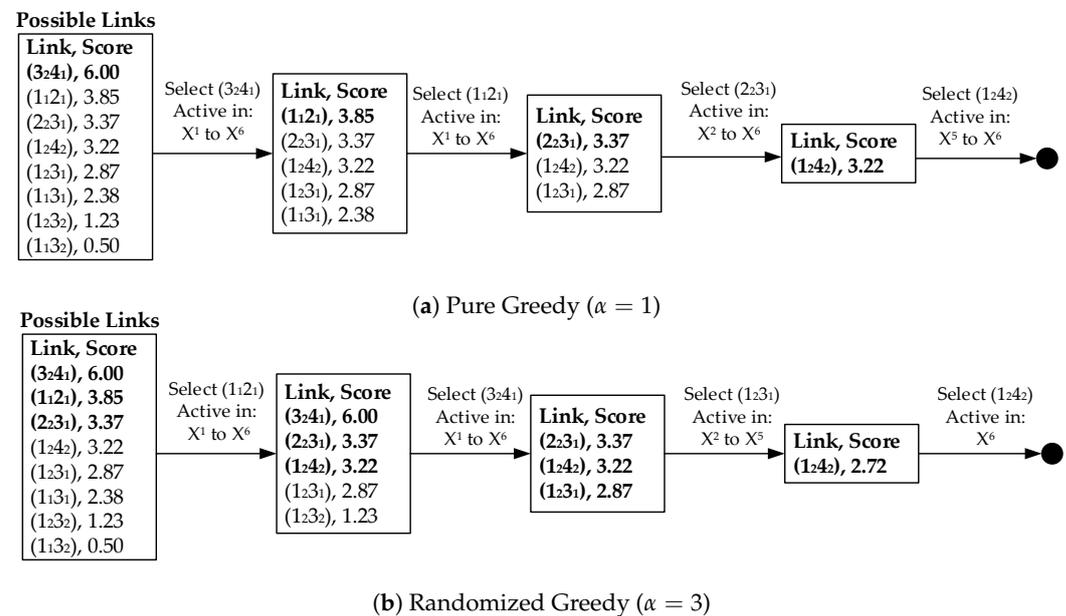


Figure 4. Examples of link selection with the Randomized Greedy Steerable Backhaul Reconfiguration Algorithm (RG-SBRA) algorithm.

In the randomized greedy example with $\alpha = 3$ shown in Figure 4b, the algorithm randomly chooses one of the 3 top scored links in each step, for example links (1_12_1) , (3_24_1) , (1_23_1) and (1_24_1) . In this scenario, intermediate link (1_23_1) is established, instead of (2_23_1) . Since interface 1_2 is part of a final link, this intermediate link can only be active from time slot $t = 2$ when it can be established, until time slot $t = 5$ as interface 1_2 needs one time slot to rotate to its final position in link (1_24_2) . Note, the score of final link (1_24_2) must be updated after step 3 as it cannot be activated until the final time slot $t = 6$.

4.2. Iter-RG-SBRA

The Iter-RG-SBRA algorithm runs several iterations of RG-SBRA with varying weights (w_1, \dots, w_7) to diversify the search. Given a set of different weight combinations \mathcal{W} , for each weight combination $\mathbf{w} \in \mathcal{W}$, the algorithm proceeds to run one iteration of RG-SBRA as pure greedy ($\alpha = 1$) and *Iter* iterations of RG-SBRA with $\alpha > 1$. The best found solution over all iterations and all weight combinations is returned as the final solution. The pseudocode of Iter-RG-SBRA is given in Algorithm 2.

Algorithm 2: The Pseudo Code of Iter-RG-SBRA

```

Input:
Problem input parameters:  $\mathcal{N}, \mathcal{M}, \mathcal{I}, \mathcal{T}, \tau, \theta, \mathbf{x}, \mathbf{L}, \mathbf{R}, \mathbf{V}, \mathbf{X}^1, \mathbf{X}^T, \mathbf{A}^1$ ;
Algorithm input parameters:  $\mathcal{W}, \alpha, Iter$ , seed
Output:  $\mathbf{X}, \mathbf{C}^+, \mathbf{C}^-, \mathbf{g}, \mathbf{r}, \mathcal{F}$ 
BestSolution  $\leftarrow \emptyset$ ;
foreach  $\mathbf{w} \in \mathcal{W}$  do
    Solution  $\leftarrow$  RG-SBRA(Problem input parameters;  $\mathbf{w}, 1$ , seed);
    if  $\mathcal{F}(\textit{Solution}) < \mathcal{F}(\textit{BestSolution})$  then
        | BestSolution  $\leftarrow$  Solution;
    end
    foreach  $k = 1, \dots, Iter$  do
        | Solution  $\leftarrow$  RG-SBRA(Problem input parameters;  $\mathbf{w}, \alpha$ , seed);
        | if  $\mathcal{F}(\textit{Solution}) < \mathcal{F}(\textit{BestSolution})$  then
        | | BestSolution  $\leftarrow$  Solution;
        | end
    end
end
return BestSolution;

```

5. Numerical Results

In this section, we evaluate the solution quality obtained by the proposed heuristic approach. To do so, we compare the following algorithms:

- **MILP:** To calculate the optimal link establishment/removal sequence that minimizes packet loss for a given reconfiguration time, we use the MILP from [14]. The MILP is run for the highest number of time slots (T) for which it could be solved, although the true optimal solution for each problem instance may be lower for higher T values.
- **Iter-RG-SBRA:** The proposed heuristic, Iter-RG-SBRA, was run for two different configurations with varying weight combinations, α values and iterations. The attribute weights were varied in values $\{0, 0.33, 0.66, 1.0\}$. In the first configuration, denoted as Iter-RG-SBRA(220), 20 random weight combinations were selected, and for each one, a pure greedy ($\alpha = 1$) iteration was run, along with 10 iterations ($Iter = 10$) with $\alpha = 10$, running RG-SBRA a total of 220 times. In the second configuration, denoted as Iter-RG-SBRA(4^7), all 4^7 weight combinations were run for a single iteration with $\alpha = 1$. Since sufficient diversification was achieved with the varying weights, further randomization with $\alpha > 1$ did not yield better results.
- **PVF-MILP:** To solve larger problem instances with higher T values in less time, we implemented a partial variable fixing heuristic in the form of a simplified MILP (denoted as PVF-MILP), which finds sub-optimal solutions by assuming final topology links are established as soon as possible. It is derived from the MILP in [14], by fixing the values of the decision variables associated with the interfaces from the final links so they start rotating towards their final positions immediately, which converts them to input parameters and, thus, significantly reduces the problem size. The remaining interfaces alignment, link establishment and traffic forwarding is then optimally solved using the MILP (with reduced complexity). This approach is described in more detail in [25] where nodes are assumed to power off when not in use.
- **DR:** Direct reconfiguration (DR) performs a straightforward transition to the target topology, where all final links are established as early as possible (as in PVF-MILP) and no temporary intermediate links are established. Initial topology links whose interfaces are not part of final topology links remain active until the end of the reconfiguration period and are then torn down. Traffic forwarding and packet loss are then calculated using the Packet Loss LP from Section 4.1.3.

All algorithms were implemented in Matlab R2017a and run on an Intel Xeon E5-2630 2.30 GHz CPU (20 cores) with 184 GB RAM (Intel, Santa Clara, CA, USA). The MILPs and LP were solved using Gurobi 7.5.2 [30].

5.1. Experimental Setup

We evaluate the proposed heuristics for 6 network scenarios created using two types of nodal distributions and a varying number of mesh backhaul nodes $|N|$, gateway nodes $|\mathcal{M}|$, and interfaces $|I|$ as outlined in Table 4. The first nodal distribution, referred to as a *Hexagon* layout and shown in Figure 1, follows a hexagonal arrangement where nodes are spaced with 140 m increments on both the x and y axis [31]. The second distribution, referred to as a *Shifted Grid*, is formed by first creating a $n \times n$ Manhattan Grid layout where all backhaul nodes are equally spaced horizontally and vertically with a spacing of 180 m [14]. The nodes are then shifted on the x and y axis by two independent random variables following a normal distribution, with $\mu = 0$ and $\sigma = \frac{\epsilon}{8}$.

Table 4. Parametrization of the network scenarios.

Network Scenario	Nodal Layout	$ \mathcal{N} $	$ \mathcal{M} $	$ I $	Users
1	Hexagon	19	1	3	105
2	Hexagon	19	1	4	105
3	Hexagon	37	2	3	210
4	Hexagon	37	2	4	210
5	Shifted Grid	25	2	3	150
6	Shifted Grid	36	3	3	210

In our model, we assume the current traffic demands congest several links in the initial topology, triggering the need for reconfiguration to a new target topology which can serve the traffic with no packet loss. To emulate this, a fixed set of users was considered for each network scenario where 70% of the users demand 50 Mbps, 20% demand 75 Mbps and 10% demand 100 Mbps. Each user demand was randomly assigned to a node in \mathcal{N} and the aggregated demands at each node n represent demand ρ_n . The target topology \mathbf{X}^T was determined by solving a simplified version of the MILP from [14] with no interface movement constraints, no initial topology input parameters and only 1 time slot. The number of users was selected such that the links were heavily loaded for $|I| = 3$ but the total demand $\sum \rho_n$ could be served by \mathbf{X}^T with zero packet loss. To create the congested initial topology \mathbf{X}^1 , the simplified MILP was solved for an alternative set of traffic demands randomly generated with the same parameters.

The maximum achievable data rate for each potential mmWave link, $R_{mm'}$, was calculated using the propagation and atmospheric path loss models given in [32] for the 60 GHz band with a transmit power level of 23 dBm. A truncated Shannon equation was employed to limit the data rate to 1–4.64 Gbps per mmWave link based on channel quality. For cases where the inter-nodal distance was such that the resulting link rates were less than 1 Gbps, it was assumed that the corresponding links could not be established, that is, the associated $L_{mm'}$ values were set to 0 (1 otherwise). The alignment angles $V_{mm'}$ were calculated based on nodal position for each distribution, with angular granularity θ set to 10° . The rotation speed of steerable antennas can be configured to a desired value in the order of tens of milliseconds per degree [24]. In this work, the time to rotate θ degrees was set to $\tau = 0.2$ s, corresponding to a 360° rotation in 7.2 s. The initial interface angular positions \mathbf{A}^1 were set according to the links active in \mathbf{X}^1 , while the initial positions of unused interfaces were selected randomly within $[0, 360]$ in multiples of θ .

While the problem does not assume a strict limit on the maximum number of reconfiguration time slots, the minimum number of time slots should allow for feasibility of the reconfiguration process depending on angular granularity and the maximum difference between initial interface positions and their final topology positions. Assuming $\theta = 10^\circ$,

$T = 19$ is the minimum number of time slots required to reconfigure a topology if at least one interface needs to rotate 180° . Consequently, the algorithms were run for T values: 19, 20, 21, 25, 30, and 35.

5.2. Results

Table 5 shows the packet loss for the solutions obtained by all described algorithm configurations for all test scenarios. Problem instances with higher T values which the MILP could not solve are denoted as ‘—’. The best solution, that is, the lowest packet loss, found for each Network Scenario is shown in bold. Note, the total packet loss in general decreases as T is increased, as a larger reconfiguration interval allows more intermediate links to be established and can distribute the tearing down of initial links over more time to achieve a smoother transition. In the case of DR, increasing the reconfiguration time does not affect the solution as all final topology link movements are initiated immediately (i.e., associated initial links are torn down simultaneously) and no intermediate links are established. For PVF-MILP, there is some improvement with the increase in T , but it is fairly limited as all final topology link movements are also initiated immediately.

We can see that for 4 out of the 6 network scenarios, Iter-RG-SBRA(4^7) was able to find better or equal solutions by increasing T than those obtained by the MILP for their highest solvable T values. Both *Iter-RG-SBRA* configurations outperformed PVF-MILP and DR significantly in all cases for all T values and can provide a good alternative for network scenarios for which the MILP cannot be solved. More specifically, for Network Scenario 1 and $T = 19$, Iter-RG-SBRA(220) and Iter-RG-SBRA(4^7) obtain solutions 15% and 10% higher than optimal, while DR gives solutions with almost 52% more and PVF-MILP, 25% more packet loss compared to the optimal. By increasing T to 35, both configurations of *Iter-RG-SBRA* were able to find solutions with lower packet loss than the MILP for the highest solvable T value. When the number of interfaces is increased to 4 (Network Scenario 2) allowing for more flexibility in link establishment, the MILP can reconfigure the backhaul at negligible loss (1 MB) for $T = 20$ and $T = 21$. The proposed greedy approaches *Iter-RG-SBRA* approximate the best MILP solution as T increases, achieving equal packet loss with Iter-RG-SBRA(4^7) for $T \geq 25$.

For Network Scenario 3, the MILP could only solve the problem for $T = 19$. In this scenario, Iter-RG-SBRA(4^7) found solutions 20% higher than the optimal (reduced to 13% by increasing T to 35), but still performed significantly better than the PVF-MILP and DR algorithms which found solutions 36% and 40% higher. When more interfaces are available (Network Scenario 4), Iter-RG-SBRA(4^7) was able to find better solutions than the MILP for $T = 35$ by almost 20%, while Iter-RG-SBRA(220), PVF-MILP and DR found solutions approx. 1%, 14%, and 66%, respectively, over that found by the MILP. For Network Scenario 5, Iter-RG-SBRA(4^7) with $T = 30$ was able to obtain a solution 4.3% superior to the best solution found by the MILP. Finally, for Network Scenario 6, Iter-RG-SBRA(4^7) gave solutions 14.5% over the best found MILP solution, but significantly outperformed PVF-MILP and DR which give solutions with 49% and 58% higher packet loss than the MILP.

Table 6 shows the execution times for all network scenarios for $T = 19, 25,$ and 35 . Note, the average execution time for only a single iteration of *RG-SBRA* is shown, as *Iter-RG-SBRA* will depend on the number of iterations run. For the optimal MILP, although the problem is bigger with an increased number of interfaces, the branch-and-bound method used by the solver could converge more quickly in scenarios with more free interfaces available and reached the optimal solution faster. While the MILP cannot be solved to optimality in reasonable time for a higher number of nodes and time slots, *RG-SBRA* is quite scalable and follows polynomial growth with complexity $\mathcal{O}(|\mathcal{N}|^2|\mathcal{I}|^2T)$. While running a single iteration of *RG-SBRA* runs less than 1s for most cases (comparable to the DR approach), running all a large number of combinations such as Iter-RG-SBRA(4^7) is in the order of hours. However, since iterations of *Iter-RG-SBRA* are independent, the total runtime of *Iter-RG-SBRA* could be significantly reduced by parallelizing its execution on multiple CPU cores. For offline planning scenarios based on traffic predictions, *Iter-RG-*

SBRA can be run with a high number of iterations providing a good alternative for network scenarios for which the MILP cannot be solved. For network scenarios for which the MILP could solve the problem for lower T values, it is recommendable to run the MILP for the highest solvable T value and Iter-RG-SBRA for larger T values and select the best solution. In an online scenario, Iter-RG-SBRA could be run for a lower number of iterations based on the given the time constraints to try to find sub-optimal solution quickly, which improve upon the fast Direct Reconfiguration approach.

Table 5. The total packet loss (in GB) for all tested algorithm configurations and network scenarios.

Network Scenario	T	MILP	Iter-RG-SBRA(220)	Iter-RG-SBRA(4 ⁷)	PVF-MILP	DR
1	19	1.256	1.446	1.392	1.572	1.908
	20	1.195	1.446	1.379	1.564	1.908
	21	1.123	1.426	1.366	1.557	1.908
	25	—	1.159	1.268	1.557	1.908
	30	—	1.139	1.108	1.557	1.908
	35	—	1.065	1.067	1.557	1.908
2	19	0.026	0.121	0.077	0.280	0.679
	20	0.001	0.116	0.014	0.280	0.679
	21	0.001	0.089	0.031	0.280	0.679
	25	—	0.051	0.001	0.280	0.679
	30	—	0.064	0.001	0.280	0.679
	35	—	0.051	0.001	0.280	0.679
3	19	2.279	2.848	2.743	3.098	3.199
	20	—	2.859	2.734	3.098	3.199
	21	—	2.934	2.728	3.098	3.199
	25	—	2.845	2.605	3.098	3.199
	30	—	2.968	2.507	3.098	3.199
	35	—	2.894	2.579	3.098	3.199
4	19	2.106	2.854	2.560	2.659	3.519
	20	—	2.841	2.529	2.645	3.519
	21	—	2.733	2.498	2.589	3.519
	25	—	2.559	2.090	2.423	3.519
	30	—	2.317	1.713	2.423	3.519
	35	—	2.130	1.693	2.423	3.519
5	19	1.915	2.032	2.005	2.045	2.05
	20	1.890	2.017	1.998	2.045	2.045
	21	1.849	2.003	1.947	2.045	2.045
	25	—	1.921	1.814	2.037	2.045
	30	—	1.875	1.769	2.037	2.045
	35	—	1.976	1.841	2.037	2.045
6	19	1.889	2.297	2.263	2.700	2.830
	20	1.792	2.326	2.239	2.700	2.830
	21	—	2.314	2.234	2.694	2.830
	25	—	2.210	2.230	2.671	2.830
	30	—	2.265	2.167	2.665	2.830
	35	—	2.213	2.054	2.665	2.830

Table 6. Execution time (in seconds).

T	Algorithm	Network Scenario					
		1	2	3	4	5	6
19	MILP	1233.64	887.23	1,647,114.47	39,841.74	41.57	3343.90
	RG-SBRA	0.12	0.15	0.43	0.50	0.22	0.40
	PVF-MILP	30.98	516.32	60.55	449.86	25.01	39.60
	DR	0.07	0.06	0.33	0.32	0.14	0.30
25	MILP	—	—	—	—	—	—
	RG-SBRA	0.17	0.20	0.63	0.71	0.25	0.60
	PVF-MILP	90.80	1099.79	127.41	1607.09	50.19	69.30
	DR	0.11	0.11	0.51	0.51	0.19	0.50
35	MILP	—	—	—	—	—	—
	RG-SBRA	0.24	0.27	1.03	1.14	0.41	0.90
	PVF-MILP	250.39	1059.70	1036.83	36,015.85	196.13	366.0
	DR	0.17	0.17	0.89	0.89	0.32	0.80

6. Conclusions

In this paper, we propose a randomized greedy heuristic to orchestrate the reconfiguration of wireless mmWave backhaul networks using mechanically-steerable antennas. The reconfiguration process involves scheduling and coordinating the rotation of the mmWave interfaces, link establishment/removal, and traffic forwarding, and should be done in such a way as to minimize packet loss during the transition. In previous work, we proposed a MILP formulation to solve the problem optimally for smaller offline instances and a limited reconfiguration time. Here, we propose a scalable heuristic algorithm to solve larger problem instances in offline planning which the MILP cannot solve, as well as a fast alternative for online planning scenarios. Results indicate that the proposed approach can provide good quality solutions in reduced time. For future work, we plan to extend the approach to optimize power consumption and use machine learning to predict target topologies based on changing traffic demands.

Author Contributions: Data curation, R.S.; Formal analysis, N.S.-K. and R.S.; Investigation, N.S.-K., R.S., H.G. and A.K.; Methodology, N.S.-K., R.S., H.G. and A.K.; Software, R.S.; Visualization, N.S.-K. and R.S.; Writing—original draft, N.S.-K.; Writing—review and editing, N.S.-K., H.G. and A.K.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the State Research Agency (Agencia Estatal de Investigación) in Spain (MINECO/AEI/FEDER, UE): project ONOFRE-2, reference number TEC2017-84423-C3-1-P. Additional funding was provided by the KK Foundation of Sweden through the SOCRA and HITS project.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Jaber, M.; Imran, M.A.; Tafazolli, R.; Tukmanov, A. 5G Backhaul Challenges and Emerging Research Directions: A Survey. *IEEE Access* **2016**, *4*, 1743–1766. [\[CrossRef\]](#)
- Feng, W.; Li, Y.; Jin, D.; Su, L.; Chen, S. Millimetre-Wave Backhaul for 5G Networks: Challenges and Solutions. *Sensors* **2016**, *16*, 892. [\[CrossRef\]](#)
- Polese, M.; Giordani, M.; Zugno, T.; Roy, A.; Goyal, S.; Castor, D.; Zorzi, M. Integrated Access and Backhaul in 5G mmWave Networks: Potential and Challenges. *IEEE Commun. Mag.* **2020**, *58*, 62–68. [\[CrossRef\]](#)
- 3GPP. *Study on Integrated Access and Backhaul (Release 16)*; Technical Report for 3rd Generation Partnership Project; 3GPP Support Office: Valbonne, France, 2018.
- Tran, G.K.; Shimodaira, H.; Rezagah, R.E.; Sakaguchi, K.; Araki, K. Practical evaluation of on-demand smallcell ON/OFF based on traffic model for 5G cellular networks. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016. [\[CrossRef\]](#)
- Mesodiakaki, A.; Zola, E.; Kassler, A. Joint User Association and Backhaul Routing for Green 5G Mesh Millimeter Wave Backhaul Networks. In Proceedings of the 20th ACM MSWiM, Miami, FL, USA, 21–25 November 2017; ACM: New York, NY, USA, 2017.
- Arribas, E.; Fernández Anta, A.; Kowalski, D.R.; Mancuso, V.; Mosteiro, M.A.; Widmer, J.; Wong, P.W.H. Optimizing mmWave Wireless Backhaul Scheduling. *IEEE Trans. Mob. Comput.* **2020**, *19*, 2409–2428. [\[CrossRef\]](#)

8. Niu, Y.; Li, Y.; Jin, D.; Su, L.; Vasilakos, A.V. A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. *Wirel. Netw.* **2015**, *21*, 2657–2676. [[CrossRef](#)]
9. Artemenko, A.; Maltsev, A.; Mozharovskiy, A.; Sevastyanov, A.; Ssorin, V.; Maslennikov, R. Millimeter-Wave Electronically Steerable Integrated Lens Antennas for WLAN/WPAN Applications. *IEEE Trans. Antennas Propag.* **2013**, *61*, 1665–1671. [[CrossRef](#)]
10. Uchendu, I.; Kelly, J.R. Survey of beam steering techniques available for millimeter wave applications. *Prog. Electromagn. Res.* **2016**, *68*, 35–54. [[CrossRef](#)]
11. Mumcu, G.; Kacar, M.; Mendoza, J. Mm-Wave Beam Steering Antenna With Reduced Hardware Complexity Using Lens Antenna Subarrays. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 1603–1607. [[CrossRef](#)]
12. Ghosh, S.; Sen, D. An Inclusive Survey on Array Antenna Design for Millimeter-Wave Communications. *IEEE Access* **2019**, *7*, 83137–83161. [[CrossRef](#)]
13. Zhang, J.A.; Hay, S.; Guo, Y.J. Directional antennas for point-to-multipoint millimetre wave communications. In Proceedings of the 2016 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC), Cairns, Australia, 19–23 September 2016; pp. 204–207. [[CrossRef](#)]
14. Santos, R.; Ghazzai, H.; Kassler, A. Optimal Steerable mmWave Mesh Backhaul Reconfiguration. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018. [[CrossRef](#)]
15. Santos, R.; Skorin-Kapov, N.; Ghazzai, H.; Kassler, A. Fast Steerable Wireless Backhaul Reconfiguration. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019.
16. Ma, Z.; Li, B.; Yan, Z.; Yang, M. QoS-Oriented joint optimization of resource allocation and concurrent scheduling in 5G millimeter-wave network. *Comput. Netw.* **2020**, *166*, 106979. [[CrossRef](#)]
17. Saad, M.; Abdallah, S. On Millimeter Wave 5G Backhaul Link Scheduling. *IEEE Access* **2019**, *7*, 76448–76457. [[CrossRef](#)]
18. Chaudhari, A.; Murthy, C.S.R. Efficient dynamic relay probing and concurrent backhaul link scheduling for mmWave cellular networks. *Comput. Commun.* **2020**, *149*, 146–161. [[CrossRef](#)]
19. Eslami Rasekh, M.; Guo, D.; Madhow, U. Joint Routing and Resource Allocation for Millimeter Wave Picocellular Backhaul. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 783–794. [[CrossRef](#)]
20. Madapatha, C.; Makki, B.; Fang, C.; Teyeb, O.; Dahlman, E.; Alouini, M.S.; Svensson, T. On Integrated Access and Backhaul Networks: Current Status and Potentials. *IEEE Open J. Commun. Soc.* **2020**, *1*, 1374–1389. [[CrossRef](#)]
21. Polese, M.; Giordani, M.; Roy, A.; Castor, D.; Zorzi, M. Distributed Path Selection Strategies for Integrated Access and Backhaul at mmWaves. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018. [[CrossRef](#)]
22. Islam, M.N.; Abedini, N.; Hampel, G.; Subramanian, S.; Li, J. Investigation of performance in integrated access and backhaul networks. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018. [[CrossRef](#)]
23. Libunao, M.; Ng, B.; Seah, W.K.G. Autonomic link management in wireless backhaul networks with OpenFlow and traffic prediction. In Proceedings of the 2017 IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, China, 22–24 October 2017. [[CrossRef](#)]
24. Santos, R.; Koslowski, K.; Daube, J.; Ghazzai, H.; Kassler, A.; Sakaguchi, K.; Haustein, T. mmWave Backhaul Testbed Configurability Using Software-Defined Networking. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 1–24. [[CrossRef](#)]
25. Santos, R.; Skorin-Kapov, N.; Ghazzai, H.; Kassler, A. Towards the Optimal Orchestration of Steerable mmWave Backhaul Reconfiguration. *Comput. Netw.* **2021**, submitted.
26. Nitsche, T.; Flores, A.B.; Knightly, E.W.; Widmer, J. Steering with eyes closed: Mm-Wave beam steering without in-band measurement. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2015), Hong Kong, China, 26 April–1 May 2015. [[CrossRef](#)]
27. Gures, E.; Shayea, I.; Alhammadi, A.; Ergen, M.; Mohamad, H. A Comprehensive Survey on Mobility Management in 5G Heterogeneous Networks: Architectures, Challenges and Solutions. *IEEE Access* **2020**, *8*, 195883–195913. [[CrossRef](#)]
28. Agyapong, P.K.; Iwamura, M.; Staehle, D.; Kiess, W.; Benjebbour, A. Design considerations for a 5G network architecture. *IEEE Commun. Mag.* **2014**, *52*, 65–75. [[CrossRef](#)]
29. Santos, R.; Kassler, A. A SDN controller architecture for Small Cell Wireless Backhaul using a LTE Control Channel. In Proceedings of the IEEE WoWMoM 2016, Coimbra, Portugal, 21–24 June 2016.
30. Gurobi Optimization. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Houston, TX, USA, 2019.
31. 3GPP. *Study on Small Cell Enhancements for E-UTRA and E-UTRAN; Higher Layer Aspects (Release 12)*. Technical Report for 3rd Generation Partnership Project; 3GPP Support Office: Valbonne, France, 2013.
32. Mesodiakaki, A.; Kassler, A.; Zola, E.; Ferndahl, M.; Cai, T. Energy efficient line-of-sight millimeter wave small cell backhaul: 60, 70, 80 or 140 GHz? In Proceedings of the IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, 21–24 June 2016. [[CrossRef](#)]