A Genetic Algorithm Approach to Virtual Topology Design for Multi-Layer Communication Networks

Uwe Bauknecht
Institute of Communication Networks and
Computer Engineering (IKR), University of Stuttgart
Stuttgart, Germany
uwe.bauknecht@ikr.uni-stuttgart.de

ABSTRACT

The core networks of current telecommunication infrastructures are typically engineered as multi-layer networks. The uppermost layer is defined by the virtual topology, which determines the logical connections between core network routers. This topology is realized by optical paths in the lower layer, which is defined by optical fiber connections between network nodes. Minimizing the hardware cost incurred by these optical paths for a given set of traffic demands is a common combinatorial optimization problem in network planning, often approached by Mixed-Integer Linear Programming. However, increasing network densities and the introduction of additional constraints will impact tractability of future network problems.

In order to provide a more scalable method, we suggest a Genetic Algorithm-based approach that optimizes the virtual topology and subsequently derives the remaining parameters from it. Our genetic encoding utilizes a combination of spanning trees and augmentation links to quickly form meaningful topologies. We compare the results of our approach to known linear programming solutions in simple scenarios and to a competing heuristic based on Simulated Annealing in large-scale problems.

CCS CONCEPTS

• **Networks** → **Network design and planning algorithms**; • **Theory of computation** → Evolutionary algorithms; • **Computing methodologies** → **Genetic algorithms**.

KEYWORDS

Genetic Algorithm, Multi-Layer Network, Virtual Topology Design

ACM Reference Format:
Uwe Bauknecht. 2021. A Genetic Algorithm Approach to Virtual Topology Design for Multi-Layer Communication Networks. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459379

1 INTRODUCTION

Internet Service Providers (ISPs) face several changes that will significantly impact future network planning. New technologies like 5G and Fiber-To-The-x (FTTx) increase data rates and enable high-reliability and low-latency services. Such Quality of Service (QoS) capabilities, coupled with a growing customer interest in novel interactive use cases, ranging from cloud-based gaming to remote-operated vehicles in logistics, require ISPs to explicitly consider QoS parameters in the design of their networks.

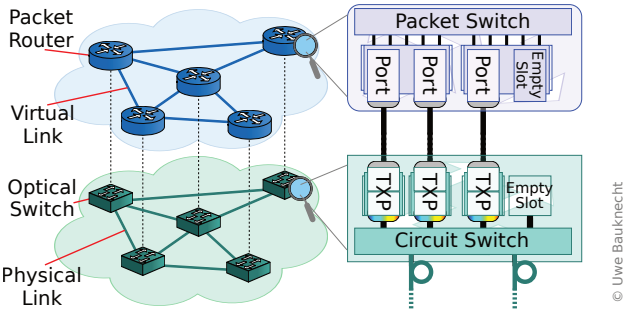

Figure 1: Multi-Layer Network Example

ISP networks typically consist of three sections. Customers connect in the access section, while the aggregation section collects the data from diverse locations, handing it to the packet routers in the core. Core networks form country-wide backbones, which feature nodes at central locations such as big cities. These core nodes are connected by optical fibers, which form the physical links of the *physical topology*, which is illustrated in Figure 1 in green color. The routers are connected by optical circuits, which are essentially high-speed laser connections routed through the optical fibers. A circuit is terminated on both sides by a Transponder (TXP), which translates the electric signal of a router port to an optical signal. Circuits can be switched between fibers such that they can traverse a path of several links. The resulting connections between the routers form the *virtual topology*, shown in blue color in Figure 1.

Both, router ports and TXPs are expensive pieces of hardware, typically in the six-figure range in US Dollars. This motivates ISPs to plan their networks such that they offer sufficient transmission capacity with the lowest amount of hardware possible. To this end, traffic may be groomed, i. e. routed on potentially longer paths if this allows using already available TXP capacity, rather than installing new TXPs. However, considering QoS requirements can prohibit grooming, since longer paths incur more transmission delay and reduce availability. Together with increasing core network densities, this results in a significant increase in problem complexity, while

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8350-9/21/07…$15.00
https://doi.org/10.1145/3449639.3459379

the desired planning time frames decrease due to lower service lead times. This can be prohibitive to established planning approaches, motivating the development of more scalable heuristics, which can provide meaningful results in short time frames.

We will explore the problem in more detail in the next section and relate it to other problems and solution approaches in Section 3. We present our contribution, consisting primarily of a problem-specific encoding and operators in Section 4 and provide experimental results based on a reference problem of reduced complexity and a real network instance in Section 5.

2 MULTI-LAYER NETWORK PLANNING

Our multi-layer network planning problems essentially consists of four sub-problems. The Traffic Demand Routing problem assigns paths of virtual links to each traffic demand, such that their QoS requirements are fulfilled. The Virtual Topology Design problem determines a topology, such that all traffic demands can be routed and the topology can be realized by circuits. The Routing and Wavelength Assignment problem assigns circuit paths to virtual links using physical links, such that the virtual topology can be created and that the capacity of a physical link is not exceeded. The Physical Topology Design problem determines a topology based on the physical link candidates, such that all circuits can be routed. The length of a physical link determines its incurred delay and availability, which are needed to determine whether the QoS constraints can be met. The interplay between these problems is illustrated in Figure 2, where blue layers relate to the virtual and green layers to the physical topology aspects, while gray layers contain the inputs to the problem.

More formally, input to our multi-layer network planning problem are the infrastructure nodes as set of vertices V, a set of potential physical edge candidates E_p and a set of traffic demands T. The edges $\langle s, d, b\rangle \in E_p$ with $s, d \in V$ and $s \neq d$ with length b in km form a simple, undirected graph $G_p = \langle V, E_p\rangle$. This represents the infrastructure of the physical layer. A traffic demand $t \in T$ is a tuple $\langle s, d, r_t, l, a\rangle$ with source and destination vertices $s, d \in V$, required data rate r_t in Gbit/s and two QoS parameters: The maximum latency l in ms and the minimum availability a in %. When no QoS is considered, i. e. all demands in T are of the form $\langle s, d, r_t, \infty, 0\rangle$, we assume that there is one demand per node pair and direction. For QoS-enabled cases, we assume one additional traffic demand between each node pair for every unique combination of QoS requirements. Furthermore, we treat all demands as indivisible aggregates.

The virtual topology also forms an undirected, simple Graph $G_v = \langle V, E_v\rangle$. Each $e_v \in E_v$ needs to provide at least as much capacity, as the sum of all r_t of the demands traversing it. This capacity is provided by the sum of the data rates of all optical circuits that realize a particular e_v. Therefore, every virtual link $e_v = \langle s, d\rangle$ needs one or more circuits with s as the first and d as the last node in a path of physical edges they traverse. Consequently, the optical circuits $c \in C$ are tuples $\langle \langle e_p\rangle_i, r_c, w\rangle$, where $\langle e_p\rangle_i$ is the path of physical links, r_c is the provided data rate in Gbit/s, and w is a wavelength slot. Several optical circuits can exist in parallel within a single physical link, as long as they have different wavelength slots.

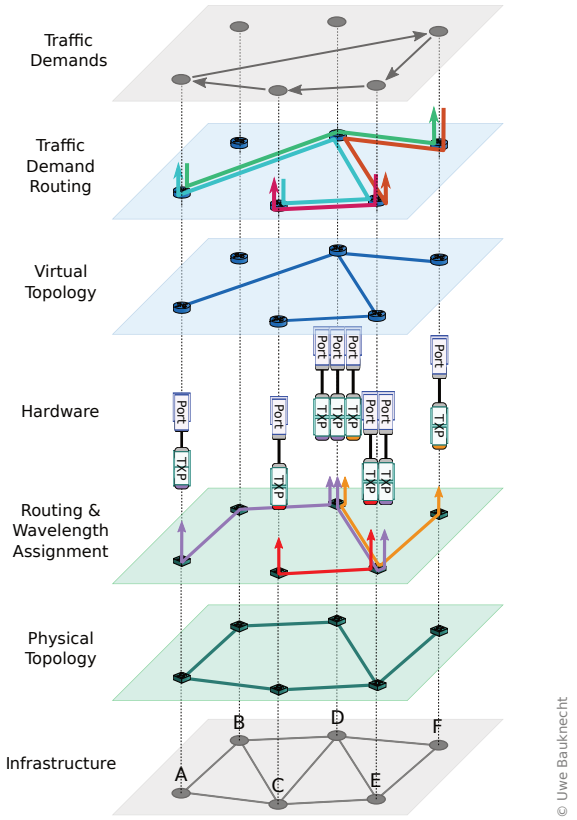

Figure 2: Multi-Layer Planning Subproblems

We also consider several technologically motivated constraints. All e_p provide an identical limited number of wavelength slots. Furthermore, the maximum attainable data rate of a circuit depends on the length of fiber, i. e. the sum of the lengths of the traversed physical edges. Modern TXP can adjust the data rate in a number of discrete steps, but every step has an individual upper reach in km. Lastly, we consider each circuit to use exactly one TXP at each end, which is connected to a port in the router. Consequently, we denote the hardware cost with a uniform cost value κ representing these devices. In our approach, we consider the primary optimization objective of minimizing the overall cost by using the least number of devices that is sufficient to provide topology and routing solutions such that all demands and constraints are satisfied.

3 RELATED WORK

Multi-layer network problems have been addressed by various methods in the past. Many scientific works utilize Mixed Integer Linear Programming (MILP) approaches, often based on extensions of minimum-cost Multi-Commodity Flow (mmMCF) formulations. The survey by Rožić *et al.* [20] lists over a dozen works solving different multi-layer problems using MILP approaches. However, large problem instances or additional hardware or QoS constraints typically require problem relaxations [22] or hybridizing MILP with other algorithms [8], such that the usage of heuristics presents a meaningful alternative.

Genetic Algorithms (GAs) have been employed to solve both, Virtual Topology Design in isolation as well as integrated within multi-layer problems. Two common approaches for topology problems are encoding link candidates using a sequence of binary selector genes or encoding a system of link weights as a real-valued gene vector, which is translated into a topology by graph algorithms. The survey of Kampstra *et al.* [15] lists over 50 works applying Evolutionary and Genetic Algorithms to such problems and an additional 14 works that specifically deal with optical networks.

A smaller number of works solve multi-layer problems somewhat similar to our approach based on applying GAs to Virtual Topology Design. Saha *et al.* [21] assume the hardware to be a fixed input and encode a connection matrix between the given TXPs from which the topology can be determined. The remaining subproblem solutions are derived by ancillary algorithms. The goal of their optimization is either to maximize throughput or minimize traffic delay. Ahmad *et al.* [1] encode a connection matrix containing the point-to-point transmission capacity for each virtual link. Their goal is to minimize the power consumption of the required hardware. Durán Barros *et al.* [10] encode a priority queue of all possible node pairs. Based on this, they route circuits on a number of precomputed paths, and run a postprocessing function to ensure connectivity. Their goal is the minimization of end-to-end delays.

Most similar to the problem presented in Section 2 is the work by Balasubramanian *et al.* [2]. They also minimize hardware cost and they consider QoS for their traffic demands. However, they focus on a different aspect of improving availability. Rather than considering it as a percentage, they specify whether and how the traffic should be treated, e. g. that it requires a preplanned backup route. Most interestingly, their network example is also rather large at over 100 nodes, but the level of density of the virtual topology is not explicitly stated. They seem to use a 2-part genetic code that consists of a binary part, representing the connection matrix, and an integer part, which represents an index to a set of precomputed routes. However, their work focuses more on the network results, rather than how their GA operates.

While the works of Balasubramanian *et al.* and Ahmad *et al.* use roughly the same optimization goal as our approach, their specific problems are different from ours in the treatment of QoS and the methodical approach. Furthermore, their focus is less on aspects of scalability, which is a gap we aim to fill with this paper.

4 GENETIC ALGORITHM APPROACH

We optimize the Virtual Topology using a generational GA with stochastic tournament replacement and derive all other solution parameters from its results. While a routing-based optimization approach offers a finer level of granularity and can therefore theoretically achieve better results, the drawback is a drastically larger search space which will inevitably limit scalability. We therefore solve the routing problems by traditional shortest-path algorithms, based on the optimized virtual topology and a full physical topology.

4.1 Encoding

We use two different encodings of the virtual topology to compare their efficacy and performance. Both approaches start from the physical infrastructure and the technological constraints by determining the maximum feasible virtual topology. This is done by computing shortest paths in terms of total length between all node pairs in the physical topology and excluding all virtual link candidates that cannot be created by a single circuit due to reach limitations of the TXPs. The first approach then encodes each feasible virtual link as a binary value, such that a chromosome for a graph of $|V|$ nodes has at most $\frac{|V|\cdot(|V|-1))}{2}$ genes. This will be referred to as Virtual Topology Binary (VTB) encoding.

A major drawback of VTB is the fact that it can represent undesirable topologies. In actual core networks, traffic routes between all node pairs are required, such that any unconnected graph can never be an optimal solution. We therefore implement a repair function that ensures graph connectivity. To this end, we initially compute an adjacency matrix for the feasible virtual topology, which indicates not just the presence, but also the position in the chromosome for each potential link. From this, we can run a connectivity check without the need to build the graph data structure first. The repair function, also uses this matrix, randomly drawing from the edges of unconnected nodes and setting the corresponding gene values.

The second method, we name Virtual Topology Centralized Spanning Tree (VTCS) encoding, is motivated by the idea of making it impossible to represent any unconnected graphs altogether by using a more complex decoding function. Inherent to any spanning tree is the property of being a connected graph featuring the least amount of edges. As such they have often been used in optimizing topology design problems, even in conjunction with GAs (e. g. [5, 9, 23]). However, trying to represent such trees in genetic codes may suffer from problems in locality and heritability [14] or increased runtimes of decoding functions [19]. Moreover, a spanning tree itself is rarely a good solution for a communication network, since it has no redundancies and can lead to very long paths as even a daisy chain topology is a valid spanning tree.

VTCS therefore needs to augment a spanning tree with additional edges to form a good topology candidate. Furthermore, it needs to reduce the number of spanning trees, since for a graph of $|V|$ nodes there will be $|V|^{|V|-2}$ different spanning trees [4], including various daisy chain topologies. The first requirement can be solved by providing a compound chromosome where some genes represent the base tree and others represent the augmentation edges. The second requirement is less easy to realize, since it is a priori difficult to assess whether a given spanning tree is a good candidate. We address this issue using a heuristic.

In generating the tree, we try to prioritize adding specifically edges that are more likely to enable shorter paths. To this end, we rank edges based on their betweenness centrality in the feasible graph, which reflects the number of shortest paths passing through the edge. To establish a strict total order relation, we additionally use a unique identifier to order edges of identical betweenness. We also determine the node with the highest nodal degree in the feasible virtual topology to be the starting node of all trees. Finally, we generate a routing table, which contains the next edge on the shortest path from any node to any other. All of these steps are performed in a preprocessing step prior to the actual GA.

The chromosome is split into two parts as shown in Figure 3. The first part uses $|V|-2$ genes to identify the spanning tree. Beginning from the starting node, the genes indicate, which node is the next

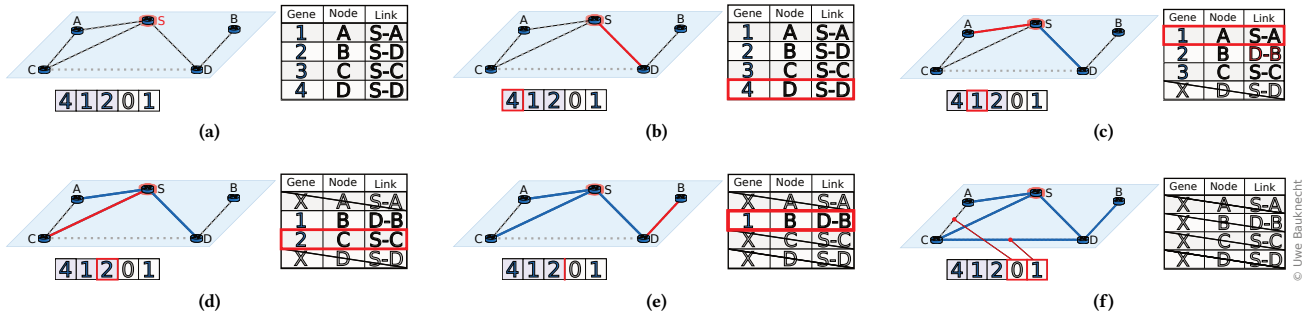

Figure 3: Decoding of a VTCS Chromosome

to be added to the spanning tree. To track the nodes to be added, we create a next-edge table, which we initialize with all those entries from the routing table that contain the starting node as shown in Figure 3a. The first gene has a range of 1 to $|V| - 1$ integer values, representing the other nodes. Using the gene value, we determine the corresponding node and the appropriate edge from the next-edge table and add them to the proto tree as visualized in Figure 3b.

Before moving on to the next gene, we have to update the next-edge table. First, we remove the row of the selected node and edge. We check the routing table for all entries with this selected node and compare them to those currently in the next-edge table. We consider only the ones connecting to nodes that have not yet been included in the tree. We replace the old entry if the selected node was also the target node of the old entry or if it has a higher value in the proposed centrality ranking. This is visible in Figure 3c.

The next gene now features $|V| - 2$ integer values, corresponding to the remaining nodes and we can repeat the process. The last among the tree genes has only 2 choices and therefore determines both, the penultimate and ultimate node, which complete the spanning tree as shown in Figure 3e. The remaining unused edges are then encoded as binary genes, as they would with VTB, which is illustrated in Figure 3f. This procedure yields at most $\frac{|V|\cdot|V|-1}{2} - (|V| - 1)$ additional genes. The exact decoding function is shown in Algorithm 1.

While the VTCS encoding promises a reduction of the search space towards meaningful solutions, it has four major drawbacks. First, there is the complexity of the decoding operation itself. While the routing table can initially be precomputed, it will still need to compare on the order of $n \log n$ entries between the tables. Second, as this procedure does not allow all spanning trees to be represented, it cannot be guaranteed that the optimum topology can be found. While this will become problematic for scarce traffic matrices, a core network with a full any-to-any traffic matrix can be expected to require a more dense topology, such that the selected trees should provide a reasonably good starting point. Third, there is an overrepresentation of the search space. By adding an augmentation link to a spanning tree of n_e edges, the resulting topology could also be achieved by n_e different spanning trees, which had originally included this edge, but their respective augmentation links were part of the original spanning tree. Ultimately, this results in an inflation factor of about $\frac{(|V|-1)!}{2^{|V|-1}}$.

Finally, since there is no fixed association between genes and links, i. e. the same gene location can refer to different links in the different chromosomes, the encoding suffers from reduced locality and heritability. This latter drawback, however, can be remediated by an appropriate operator.

4.2 Operators

As standard operators, we consider a 3-Point Crossover (3PXO) and the Random Reset Mutation (RRM). We also consider a Creep mutation that simply increments or decrements an integer value at equal probability by a value of i, wrapping between lowest and highest values. These will be used to perform baseline tests. Alternatively, we utilize the following three more problem-specific operators, designed to enhance the performance for the proposed encodings. For the VTB encoding, we suggest Link Block Crossover (LBXO). Rather than transferring contiguous gene sequences between chromosomes, LBXO randomly chooses a variable number of nodes in the graph and exchanges those genes that attach to these specific nodes. This can facilitate grooming since it helps maintain the routing at that node. Otherwise, changing many links at one node could completely alter the routes there. The number of genes to be exchanged can also be scaled according to the cost difference between the individuals, such that a low-quality solution will receive more genes from a high-quality solution.

Due to their mixture of integer and binary chromosomes, we also suggest specific operators for the VTCS encoding. The first is the VTCS-Specific Mutation (VSM) operator. Since the tree genes are relatively few, they are not very likely to be changed. However, if they are affected the solution will change drastically. To control these effects, VSM has a parameter to adjust the probability that a tree gene or link gene should be mutated. After deciding this, a gene from the respective substring is selected at random and manipulated in the same way as by Creep mutation with increment size $i = 1$.

Finally, we also suggest a VTCS-Specific Crossover (VSXO) operator. Since the gene sequences in identical places of two different VTCS chromosomes may represent very different links, a regular crossover operator will result in low heritability of traits. To counteract this, VSXO first takes the tree genes of the parents and copies them without change to the offspring. Now it can determine which genes correspond to which links in the offspring. It then

Algorithm 1 Spanning Tree Encoding – Decoding Function

Require: $\langle V, E_v \rangle$ is a connected, non-empty graph
Require: *edges* provides feasible edges as priority-sorted array
Require: *genes* provides gene sequence as array
function DECODE(V,E_v,*edges*,*edgeMap*, v_{start}, *genes*)
let $tmpEdges \leftarrow edges$
let $V_{tmp} \leftarrow V \setminus \{v_{start}\}$
let $i_{target} \leftarrow 0,\ i \leftarrow 0,\ v_{curr} \leftarrow v_{start}$
let $nextEdgeMap \leftarrow edgeMap[v_{start}]$
let $E_{tmp} \leftarrow \emptyset$
for $i_{gene} = 0$ **to** $i_{gene} = |V| - 2 - 1$ **do**
$i \leftarrow 0$
$i_{target} \leftarrow genes[i_{gene}]$
$nextEdgeMap[v_{curr}] \leftarrow \perp$
for all $\langle v, \langle s, d \rangle\rangle \in nextEdgeMap$ **do**
if $i \leq i_{target}$ **then**
$v_{curr} \leftarrow d$
if $v_{curr} \in V_{tmp}$ **and** $\exists \langle n, \langle s, d \rangle\rangle \in tmpEdges$ **then**
if $i = i_{target}$ **then**
$V_{tmp} \leftarrow V_{tmp} \setminus \{v_{curr}\}$
$tmpEdges[n] \leftarrow \perp$
$\exists! \langle m, \langle d, s \rangle\rangle \in tmpEdges$
$tmpEdges[m] \leftarrow \perp$
$E_{tmp} \leftarrow E_{tmp} \cup \{\langle s, d \rangle, \langle d, s \rangle\}$
end if
$i \leftarrow i + 1$
end if
end if
end for
for all $v \in V_{tmp}$ **do** ▷ Update Next-Edge Table
$\langle s, d \rangle \leftarrow edgeMap[v_{curr}][v]$
if $d \in V_{tmp}$ **and** $\exists \langle n, \langle s, d \rangle\rangle \in tmpEdges$ **and** $\hookrightarrow nextEdgeMap[v] \neq \perp$ **then**
$\langle o, t \rangle \leftarrow nextEdgeMap[v]$
if $(t \neq v$ **and** $d = v)$ **or** $v_{curr} = t$ **or** $\hookrightarrow (\langle o, t \rangle \succeq_E \langle s, d \rangle$ **and** $t \neq v)$ **then**
$nextEdgeMap[v] \leftarrow \langle s, d \rangle$
end if
end if
end for
end for
$\exists! v \in V_{tmp}$ ▷ Only exactly one element remaining
$v_{curr} \leftarrow v$
$\langle s, d \rangle \leftarrow nextEdgeMap[v_{curr}]$
$\exists! \langle n, \langle s, d \rangle\rangle \in tmpEdges$
$tmpEdges[n] \leftarrow \perp$
$\exists! \langle m, \langle d, s \rangle\rangle \in tmpEdges$
$tmpEdges[m] \leftarrow \perp$
$E_{tmp} \leftarrow E_{tmp} \cup \{\langle s, d \rangle, \langle d, s \rangle\}$
for all $e \neq \perp \in tmpEdges$ **do**
$i_{gene} \leftarrow i_{gene} + 1$
if $genes[i_{gene}] = 1$ **then**
$E_{tmp} \leftarrow E_{tmp} \cup \{e\}$
end if
end for
return E_{tmp}
end function

activates the augmentation links corresponding to the opposite chromosome's tree links. The last step is to exchange the remaining links based on a regular probabilistic crossover.

4.3 Population Management

Since our goal is to develop a single good solution as quickly as possible, we can expect a meaningful initialization to accelerate the search. Following the notion that a link-augmented spanning tree provides a good starting point, we propose to initialize the population for VTB chromosomes with just such trees. To this end, we adapt the repair function that ensures connectivity for VTB and add further augmentation links according to a fixed ratio. We found in experiments that for our scenarios a relatively high ratio of 0.7 was most useful. We will refer to this initialization as Augmented Spanning Tree Initialization (ASTI). Note, that randomly initializing a VTCS chromosome with this probability will have the same effect as ASTI for VTB.

Furthermore, we suggest including two trivial solutions along with those generated by ASTI. The first is the solution, where all feasible links are activated. The second activates exactly those virtual links that correspond exactly to the links of the physical topology. We refer to the combination of these trivial solutions and ASTI as *hybrid* initialization. Finally, we always ensure that no duplicate chromosomes exist in the population and that the best individual cannot by removed by the replacement process.

5 EVALUATION

We evaluate the aforementioned approaches in two different scenarios. In order to establish a meaningful baseline in attainable solution quality, we use a known problem from the SNDlib [16] database of reference networking problems. While there is no problem matching our intended level of complexity regarding QoS-enabled traffic, the problem called *france--D-B-M-N-S-A-N-N* can in fact be interpreted as a multi-layer network problem of reduced complexity. The second part of the evaluation is based on full-featured scenarios that have been modeled after actual networks with realistic traffic matrices. In this part, we focus on the ability of given methods to achieve meaningful improvements over the baseline solution within limited time frames.

For comparison, we include results from a Shortest Path Heuristic (SPH), which simply uses shortest paths in a virtual topology identical to the physical topology. Furthermore, we employ two different approaches based on Simulated Annealing. Both use the random activation or deactivation of virtual links as perturbation function. The first, which we will abbreviate as SaT, reroutes all traffic following a perturbation of the virtual topology. The second, which we refer to as SaR, reroutes only the traffic demands affected by the change. SaR can therefore compute an iteration faster than SaT and it can passively optimize routing. These approaches are based on the original works of Feller [12, 13]. We start both approaches from a virtual topology, which corresponds to the physical topology, such that the result can never be worse than what SPH can achieve.

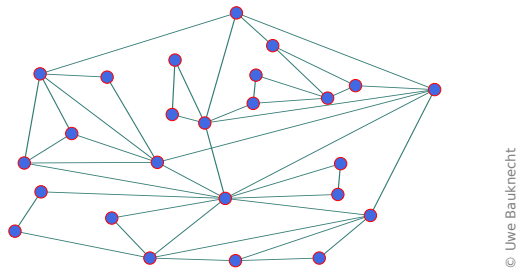

Figure 4: Topology of *france--D-B-M-N-S-A-N-N*

5.1 SNDLib Reference Scenario

The problem *france--D-B-M-N-S-A-N-N* contains the infrastructure of a french core network with 25 nodes and 45 physical link candidates. This topology of density 0.15 is shown in Figure 4. The included traffic matrix contains 300 unidirectional demands of between 332 and 1808 required data rate units, but regrettably no QoS classes. Some terms and parameters differ slightly from those described in Section 2: Circuits cannot be optically switched and there is no limit on the number of circuits on a single physical link. Furthermore, a pair of TXPs for a circuit is referred to as a "module" and all modules have a uniform cost κ_{SND} of 250 cost units while providing a capacity of 2500 data rate units at infinite reach.

The cost function included with the problem is as follows.

$$F_{Cost} = |C| \cdot \kappa_{SND} \tag{1}$$

For our optimization algorithm, we add two more penalty terms, such that we arrive at

$$F_{Opt} = |C| \cdot \kappa_{SND} + u_d \cdot \alpha + u_c \cdot \beta \tag{2}$$

with

$$\alpha > \beta$$
$$\beta > \kappa_{SND}$$
$$\alpha + r_{t,min} \cdot \beta \gg |E_v| \cdot \kappa_{SND}$$
$$\alpha, \beta \in \mathbb{R}_+$$

where $r_{t,min}$ is the lowest rate of all demands, u_d is the number of unrouted demands and u_c is the sum of the unrouted capacity. The scaling factors α and β are chosen such that the penalty incurred for not routing the lowest data rate is still higher than that of the longest possible path.

All metaheuristic approaches were run with a time limit of 12 h and repeated 10 times with independent seeds to control for statistical effects. All tests were run on dedicated servers with dual-socket Xeon E5-2640 v4 processors, each featuring 10 cores and a maximum frequency of 3.4 GHz. The Simulated Annealing methods use different cooling schedules. While SaR was run with a cooling factor of 0.95 at 6000 iterations per temperature, SaT had to be run with a much slower 0.9999 at 10000 iterations, as it would otherwise lead to premature convergence. The GA methods were run on 20 parallel threads with the parameters listed in Table 1.

Figure 5 shows boxplots of the final value of F_{Cost} for each of the metaheuristics and the time required to reach this value. In the left sub-figure, the top line represents the cost value achieved by the SPH. Unfortunately, SNDlib's database does neither contain

Table 1: Genetic Algorithm Parameters in SNDlib Scenario

Encoding	VTB	VTCS
Population Size	10,000	10,000
Offspring	1,000	1,000
Initialization	ASTI	Random
Mutation	RRM, $p = 0.02$	Creep, $i = 1$
Recombination	3PXO	3PXO

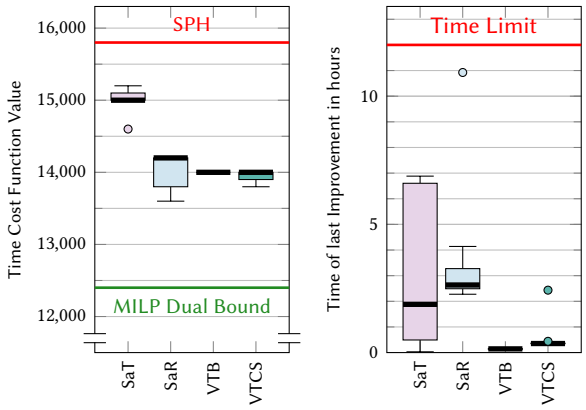

Figure 5: Cost and Runtime for SNDlib Scenario

an exact optimum, nor a primal solution, likely due to problem complexity. However, a dual bound, i. e. a lower bound to the actual optimum obtained from a MILP approach by relaxation, is present and indicated by the lower line in the left figure. All metaheuristics achieve better cost values than the simple SPH. This meets expectations since SPH cannot enforce grooming by eliminating underutilized links. Furthermore, both Simulated Annealing methods use the SPH solution as a starting point. While the SaT approach lags behind, the overall best result was achieved by the SaR approach, which is even better than all results obtained from the GA methods, probably due to its ability to influence the traffic routing more directly. However, VTB and VTCS show better median values and VTCS has the best average cost value.

When comparing the times until these values were reached, the GA approaches drastically outperform SaT and SaR, thanks to their parallel implementation with all VTB runs being below 10 min, while VTCS took less than 30 min, except for one outlier. When comparing the number of candidate solutions evaluated during runtime, such that the speedup due to parallel implementation is irrelevant, we observe that SaR requires about 5 % fewer evaluations than VTB, while VTCS needed almost 63 % fewer, when ignoring the outlier.

5.2 QoS Scenario

To analyze our approaches in scenarios covering all technological and QoS constraints, we collected data about the network of a North American ISP and reverse-engineered the physical link lengths by

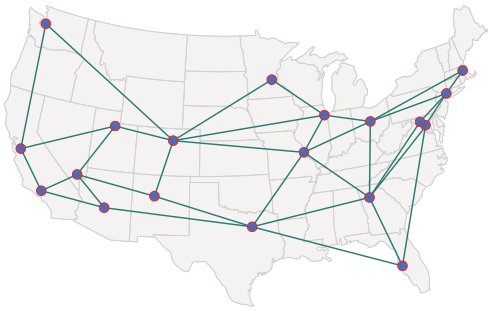

Figure 6: Topology of USA-DC

Table 2: TXP Transmission Modes

Data Rate in Gbit/s	50	100	150	200	250
Reach in km	8,000	4,000	2,000	1,000	500
V. Link Coverage in %	100	92.4	50.3	18.1	5.3

correlating this data with road and rail networks. We refer to this topology as *USA-DC*. It consists of 19 nodes and 35 links as shown in Figure 6. For the TXPs, we assume 5 different transmission modes, based on data obtained from technical documentation [6, 7]. We assume that the mode of highest data rate for a given link length is used. The modes and the resulting coverage of potential virtual links are shown in Table 2. Since 100 % of these links can be realized, we need to consider the full graph of 171 virtual links.

Traffic data was obtained from a generator developed by Enderle and the author [11] using public forecasts from Cisco [17, 18] and US Census surveys [3]. There are 812 unidirectional demands with a sum data rate of 47.4 Tbit/s. Among these, 23.3 % require a maximum latency of 10 ms and 17.4 % a minimum availability of 0.99 %. We determine the incurred traffic delay for each link, based on the speed of light in optical fiber at about 4.8985 µs/km. Additionally, we consider 1 ms per router due to packet processing and queuing. Availability is modeled to decrease with fiber length, since longer distances increase the risk of a fiber defect, e. g. due to a digging accident. We assume a figure of 2.55×10^{-6} km^{-1}, resulting in an availability of 0.99745 for a distance of 1000 km.

For the optimization cost function in Equation 3, we consider the TXPs at a unit cost of $\kappa = 1$, while also including penalty terms for unfulfilled QoS requirements.

$$F_{QoS} = |C| \cdot 2 \cdot \kappa + u_d \cdot \phi + u_a \cdot \psi + u_l \cdot \omega \tag{3}$$

with

$$\phi > |V| \cdot \frac{r_{t,max}}{r_{c,min}} \cdot \kappa$$
$$\kappa > |T| \cdot \psi$$
$$\psi > |T| \cdot \omega$$
$$\kappa, \phi, \psi, \omega \in \mathbb{R}_+$$

where $r_{t,max}$ is the maximum data rate of all demands in T and $r_{c,min}$ is the minimum data rate of all TXP modes. Small greek letters indicate weighting parameters. The variable u_d represents the number of unrouted demands, u_a represents the number of demands failing their availability requirement, whereas u_l is the number of demands that exceed their latency limit.

Table 3: Genetic Algorithm Parameters in QoS Scenario

Version	VTB-R	VTB-E	VTCS-R	VTCS-E
Encoding	VTB	VTB	VTCS	VTCS
Population Size	400	400	400	400
Offspring	100	100	100	100
Initialization	Random	Hybrid	Random	Hybrid
Mutation	RRM, $p = 0.02$	RRM, $p = 0.02$	Creep, $i = 1$	VSM, $p = 0.15$
Recombination	3PXO	LBXO	3PXO	VSXO

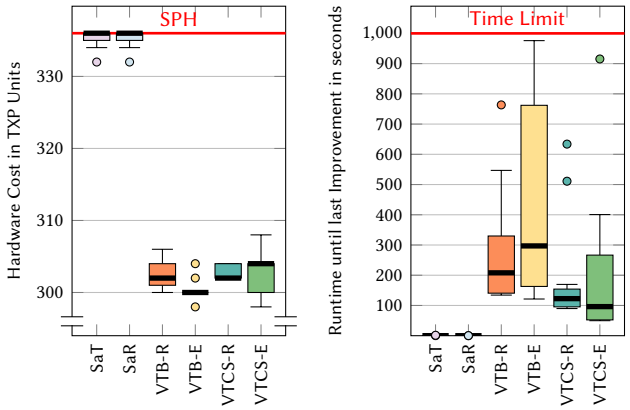

Figure 7: Cost and Runtime for USA-DC Scenario

In this evaluation, we focus on the potential to achieve improvements over SPH in short time frames. The runtime limit is set to a mere 1000 seconds. All proposed approaches are used and the algorithmic parameters adjusted to match this scenario. For Simulated Annealing, this means compressing the cooling schedules, such that a meaningful amount of temperature steps towards refinement can be taken in the available time. We set the cooling factors for SaT and SaR to 0.95 and reduce the number of iterations per temperature to 1000. For the GA approaches, we also reduce the population and offspring sizes to allow for a meaningful number of generations. We use two different versions per encoding. The versions abbreviated VTB-R and VTCS-R use the regular operators and random initialization, while VTB-E and VTCS-E use enhanced operators and initialization schemes as listed in Table 3.

While all obtained solutions succeed in fulfilling all QoS requirements, the resulting hardware costs vary considerably as presented by the boxplots in Figure 7. The drastically increased complexity in conjunction with the limited timeframe compared to the previous problem has a large impact on the Simulated Annealing methods. Their median values remain identical to the SPH solution, which is their starting point. Only few runs showed a minor improvement within the initial 20 of about 80,000 iterations. The GAs can analyze

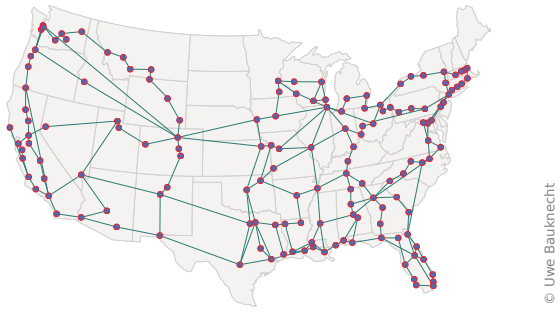

Figure 8: Topology of USA-Max

Table 4: Feasible Virtual Topology Variations of USA-Max

Version	Physical	Small	Medium	Large	Full
Nodes	149	149	149	149	149
V. Links	206	1,755	4,725	9,345	11,026
Density	0.0187	0.159	0.429	0.848	1

more than 500,000 solutions in the same time thanks to their parallel implementation. They improve the cost value by between 8 and 11 % compared to SPH, with both enhanced versions achieving the best overall value at 298. While VTCS-E exhibits the lowest median time, this comes at the drawback of being much less consistent in cost and time values, such that the median cost value for VTCS-R is lower than for VTCS-E.

5.3 Large-Scale QoS Scenario

The final scenario is geared towards exploring the scalability limits of the proposed approaches. Based on the data sources in the previous scenario, we have also created a much larger topology including all known sites of the ISP. It is referred to as *USA-Max* and consists of 149 nodes and 206 links as shown in Figure 8. All virtual links are feasible assuming the transmission modes listed in Table 2, which results in a full topology of 11,026 virtual links. In order to approach this level of complexity more gradually, we chose to create several intermediate feasible topologies as shown in Table 4 by artificially limiting the reach of the TXP modes. The traffic generator from the last scenario is used once more to generate the traffic matrix. The sum data rate of the resulting 47,094 traffic demands amounts to 44.8 Tbit/s with 22.9 % requiring the previous latency and 19.2 % the previous availability requirements.

We focus on the fastest of the previous approaches. For Simulated Annealing, we will use only SaR, and for the GA approaches VTB-E and VTCS-E. However, some adjustments are made to increase the scalability. With topologies of the current level of complexity, the supporting algorithms in traffic and especially circuit routing become a bottleneck, such that a single network cost evaluation would require on the order of minutes, which is prohibitive to the proposed optimization timeframe of 1,000 seconds. We therefore reduce the population size to 10 and the number of offspring to 20. For VTB we also omit the repair function and slightly change the hybrid initialization. Rather than creating augmented spanning

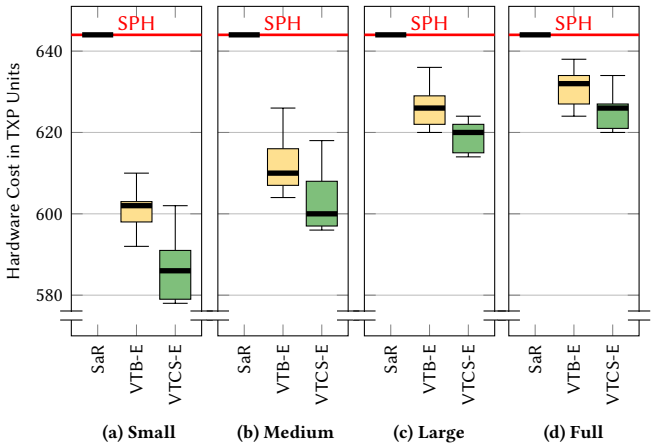

Figure 9: Cost for USA-Max Scenario Versions

trees, we use a randomly augmented physical topology to avoid the computational overhead of creating random trees. Finally, we also apply a global relaxation to the problem for all methods by lifting the wavelength continuity constraint, i. e. we skip the steps of properly assigning a wavelength slot. This is necessary, since it is the least scalable part of the supporting algorithms, ultimately consuming more than 5 times the computing time of all other algorithms combined.

Results for the different topologies is shown in Figure 9. The Simulated Annealing method was unable to achieve any improvement over the SPH baseline in all experiments. The GA versions are initially still capable of achieving an average improvement of 6.55 % for VTB and VTCS even improves the cost value by 8.7 %. However, the achievable gains rapidly shrink with growing numbers of feasible links. For the full topology this amounts to a 1.96 % improvement for VTB, while VTCS is capable of 2.89 % on average.

6 CONCLUSION

We have presented a multi-layer networking problem description, common in planning and operation for typical core networks of large ISPs. Rather than optimizing all aspects of such problems jointly, we suggest focusing on one of its primary constituent problems to provide meaningful scalability. We therefore presented an established and a novel approach to encode a virtual topology and suggested several specific adaptions to operators and initialization procedures to accelerate the optimization process.

We determined that our approaches outperform the simple baseline heuristic as expected and that they provide performance comparable to other metaheuristics, while exploiting approximately half of the headroom towards known dual bounds in reference problems. Furthermore, we could show that the cost improvements carry over to complex problems including relevant technological and QoS parameters while operating on limited time budgets. Finally, provided that the supporting algorithms retain sufficient performance, we have also shown that the approach can even scale towards large networks, albeit at decreasing solution quality.

REFERENCES

[1] Arsalan Ahmad, Andrea Bianco, Edoardo Bonetto, Davide Cuda, Guido Gavilanes Castillo, and Fabio Neri. 2011. Power-aware logical topology design heuristics in Wavelength-Routing networks. In *15th International Conference on Optical Network Design and Modeling - ONDM 2011*. 1–6.

[2] Srivatsan Balasubramanian, Satyajeet Ahuja, Gaya Nagarajan, Andrea Celletti, and Frantisek Foston. 2017. Multilayer Planning for Facebook Scale Worldwide Network. In *2017 International Conference on Optical Network Design and Modeling (ONDM)*. 1–6. https://doi.org/10.23919/ONDM.2017.7958520

[3] United States Census Bureau. 2019. *Annual Estimates of the Resident Population for Incorporated Places of 50,000 or More, Ranked by July 1, 2018 Population: April 1, 2010 to July 1, 2018*. Retrieved August 18, 2019 from https://factfinder.census.gov/bkmk/table/1.0/en/PEP/2018/PEPANNRSIP.US12A

[4] Arthur Cayley. 1897. A Theorem on Trees. In *The Collected Mathematical Papers of Arthur Cayley, Sc.D., F.R.S.* Vol. XIII. Cambridge University Press, 26–28.

[5] Hsinghua Chou, G Premkumar, and Chao-Hsien Chu. 2001. Genetic algorithms for communications network design-an empirical study of the factors that influence performance. *IEEE Transactions on Evolutionary Computation* 5, 3 (2001), 236–249.

[6] Cisco Data Sheet. 2017. *Cisco NCS 1002*. Cisco Data Sheet. https://www.cisco.com/c/en/us/products/collateral/optical-networking/network-convergence-system-1000-series/datasheet-c78-733699.pdf

[7] Cisco Data Sheet. 2018. *Cisco NCS 2000 200-Gbps Multirate DWDM Line Card Data Sheet*. Cisco Data Sheet. https://www.cisco.com/c/en/us/products/collateral/optical-networking/network-convergence-system-2000-series/datasheet-c78-733699.pdf

[8] Amaro de Sousa, Artur Tomaszewski, and Michał Pióro. 2016. Bin-packing based optimisation of EON Networks with S-BVTs. In *2016 International Conference on Optical Network Design and Modeling (ONDM)*. 1–6. https://doi.org/10.1109/ONDM.2016.7494082

[9] Der-Rong Din. 2015. Genetic algorithm for virtual topology design on MLR WDM networks. *Optical Switching and Networking* 18 (2015), 20 – 34. https://doi.org/10.1016/j.osn.2015.03.003

[10] Ramón J. Durán Barroso, Ignacio de Miguel, Noemí Merayo, Patricia Fernández, Rubén M. Lorenzo, and Evaristo J. Abril. 2009. Minimisation of end-to-end delay in reconfigurable WDM networks using genetic algorithms. *European Transactions on Telecommunications* 20, 8 (2009), 722–733. https://doi.org/10.1002/ett.1344 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.1344

[11] Tobias Enderle and Uwe Bauknecht. 2018. Modeling Dynamic Traffic Demand Behavior in Telecommunication Networks. In *Proceedings of the 19th ITG-Symposium in Photonic Networks 2018*. VDE, 18–25.

[12] Frank Feller. 2013. Evaluation of a Centralized Method for One-Step Multi-Layer Network Reconfiguration. In *Proceedings of the 24th Tyrrhenian International Workshop on Digital Communications (TIWDC 2013)*.

[13] Frank Feller. 2016. *A Reconfiguration Method for Energy-Efficient Operation of Multi-Layer Core Networks - Communication Networks and Computer Engineering Report No. 113*. Ph.D. Dissertation. Universität Stuttgart.

[14] Jens Gottlieb, Bryant A. Julstrom, Günther R. Raidl, and Franz Rothlauf. 2001. Prüfer Numbers: A Poor Representation of Spanning Trees for Evolutionary Search. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (San Francisco, California) *(GECCO'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 343–350. http://dl.acm.org/citation.cfm?id=2955239.2955297

[15] Peter Kampstra, Robert D. Van der Mei, and Agoston E. Eiben. 2006. *Evolutionary computing in telecommunication network design: A survey*. Technical Report. Vrije Universiteit, Faculty of Exact Sciences and CWI, Advanced Communication Networks. Amsterdam, Netherlands.

[16] Sebastian Orlowski, Roland Wessäly, Michał Pióro, and Artur Tomaszewski. 2010. SNDlib 1.0âĂŤSurvivable Network Design Library. *Networks* 55, 3 (5 2010), 276–286. https://doi.org/10.1002/net.20371

[17] Cisco White Paper. 2017. *Cisco Global Cloud Index: Forecast and Methodology, 2016–2021*. Technical Report.

[18] Cisco White Paper. 2017. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. Technical Report. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[19] Günther R Raidl and Bryant A Julstrom. 2003. Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on evolutionary computation* 7, 3 (2003), 225–239.

[20] Ćiril Rožić, Dimitrios Klonidis, and Ioannis Tomkos. [n.d.]. A survey of multi-layer network optimization. In *2016 International Conference on Optical Network Design and Modeling (ONDM)*.

[21] Debashis Saha, M.D. Purkayastha, and Amitava Mukherjee. 1999. An approach to wide area WDM optical network design using genetic algorithm. *Computer Communications* 22, 2 (1999), 156 – 172. https://doi.org/10.1016/S0140-3664(98)00238-2

[22] Mohammad Sheikh Zefreh, Ali Tizghadam, Alberto Leon-Garcia, Halima Elbiaze, and Walter Miron. 2016. CAPEX optimization in multi-chassis multi-rate IP/optical networks. In *2016 International Conference on Computing, Networking and Communications (ICNC)*. 1–6. https://doi.org/10.1109/ICCNC.2016.7440611

[23] Gengui Zhou and Mitsuo Gen. 1999. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research* 114, 1 (1999), 141–152.