



Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks

Jaime Galán-Jiménez¹ · Marco Polverini² · Francesco G. Lavacca² · Juan Luis Herrera¹ · Javier Berrocal¹

Received: 11 February 2022 / Accepted: 9 August 2022 / Published online: 8 September 2022
© The Author(s) 2022

Abstract

Software-defined networking (SDN) is a paradigm that provides flexibility and programmability to computer networks. By introducing SDN nodes in a legacy IP network topology, network operators can benefit on higher control over the infrastructure. However, this migration is not a fast or straightforward process. Furthermore, to provide an adequate quality of service in hybrid IP/SDN networks, the coordination of both IP and SDN paradigm is fundamental. In this paper, this coordination is used to solve two optimization problems that are typically solved separately: (i) traffic load balancing and (ii) power consumption minimization. Each of these problems has opposing objectives, and thus, their joint consideration implies striking a balance between them. Therefore, this paper proposes the Hybrid Spreading Load Algorithm (HSLA) heuristic that jointly faces the problems of balancing traffic by minimizing link utilization and network's power consumption in a hybrid IP/SDN network. HSLA is evaluated over differently sized topologies using different methods to select which nodes are migrated from IP to SDN. These evaluations reveal that alternative approaches that only address one of the objectives are outperformed by HSLA.

Keywords Hybrid IP/SDN · Load balancing · Energy efficiency · Traffic engineering · Heuristic

1 Introduction

The internet has drastically changed how our society creates, shares, and consumes information. As a consequence, the volume of traffic flowing through the internet increases exponentially over time, as the number of users continues rising from 3.9 billion in 2018 to the predicted 5.3 billion

by 2023 [1], generating an estimate of 2.6 Exabytes of traffic daily [2].

It is because of this growth in traffic demand that, to address peak traffic loads, as well as to react to events that degrade network performance, such as link failures [3] or router malfunctions [4], internet service providers (ISPs) need to reinforce the existing network infrastructure. Such an objective can be obtained by overprovisioning existing resources or by providing additional network redundancy. However, the use of such techniques leads to high energy consumption within normal network operations because all the network equipment is powered on, regardless of the traffic flowing through it [5].

Nonetheless, the flexibility and the programmability provided by the software-defined networking (SDN) paradigm, through the global network knowledge obtained by the controller, enable the proposal of solutions that optimize the amount of network equipment (e.g., links and nodes) that need to be powered on to successfully fulfill the quality of service (QoS) requirements [6–8]. These solutions adapt the network configuration to the traffic demand and improve the network's energy efficiency [9]. Furthermore, the use of SDN allows for the application of additional solutions,

✉ Jaime Galán-Jiménez
jaime@unex.es

Marco Polverini
marco.polverini@uniroma1.it

Francesco G. Lavacca
francescogiaccinto.lavacca@uniroma1.it

Juan Luis Herrera
jlherrerag@unex.es

Javier Berrocal
jberolm@unex.es

¹ Department of Computer Systems and Telematics Engineering, University of Extremadura, Avda. de la Universidad, S/N, 10003 Cáceres, Spain

² DIET Department, University of Rome “Sapienza”, Via Eudossiana, 18, 00184 Rome, Italy

such as traffic engineering (TE) techniques. SDN enables the network to design optimal policies for routing and, therefore, flow forwarding, thus performing traffic load balancing at the network level [10–12].

To make use of the SDN paradigm in a network infrastructure, the network equipment needs to be replaced with SDN-compatible nodes. However, such migration is not a straightforward or fast process, and it is not feasible to migrate large network topologies in the short term (e.g., labeling a *flag day* to migrate from IP to SDN), especially due to the economic costs of the replacement and the lack of provision of service-level agreement (SLA)-included services to the ISP's customers during the migration. A solution to this problem, commonly used by ISPs due to its cost-effectiveness, is to migrate incrementally, replacing only a subset of network elements and evaluating their practicality after a certain period of time [13]. Such partial migrations lead to hybrid IP/SDN networks, where both legacy IP nodes and SDN nodes coexist. In this type of network, both kinds of equipment need to be coordinated, as well as the protocols used for routing (i.e., OSPF and OpenFlow) [14, 15].

The problem of energy efficiency has been studied in related literature. In [16], an optimal, linear programming-based solution that optimizes the energy efficiency of an SDN network, along with a greedy heuristic for its application in large networks, is proposed. These solutions consider the possibly limited number of OpenFlow rules that can be installed in each SDN switch and provide up to 18.2% energy savings w.r.t. no energy efficiency solutions. However, this proposal is designed for pure SDN networks and is, thus, unsuitable for hybrid IP/SDN networks. Wang et al. extended the problem to hybrid IP/SDN networks in [17]. This work demonstrated the NP-hardness of the energy efficiency optimization problem in hybrid IP/SDN networks and proposed a heuristic algorithm to solve the problem based on spanning tree algorithms. The impact of traffic load, network size, and SDN nodes' number on the achieved energy savings was also analyzed.

A different approach for energy efficiency optimization is described in [18], which optimizes the energy efficiency of a hybrid IP/SDN network by proposing the SENAtOR algorithm. SENAtOR employs a *smooth power-off* technique where SDN switches stop sending control traffic to legacy IP equipment before being powered off, giving the IP part of the infrastructure time to adapt. SENAtOR is shown to have near-optimal performance and takes only milliseconds to execute. Finally, [19] presented *Hybrid Energy-Aware Traffic Engineering* (HEATE). In contrast to the previous approaches, which focused on optimizing energy efficiency on SDN devices and allowing the solution to be interoperable with IP equipment, HEATE actively optimized energy efficiency in both kinds of equipment for hybrid IP/SDN networks. To do so, HEATE found the optimal setting of

the OSPF link weight and the split ratio on SDN nodes. In particular, HEATE considered that IP routers performed shortest path routing by optimizing OSPF link weights, while SDN nodes satisfied multipath routing with traffic flow splitting through the action rules defined by the controller. HEATE is shown to provide better performance than flow allocation-based and OSPF-based energy efficiency algorithms in multiple network topologies. However, all these approaches are fundamentally different from the proposal of this work, as they exclusively focus on energy consumption, while other crucial aspects (e.g., QoS metrics, link performance) are not considered [20].

The consequence of single-objective energy efficiency optimization is that load balancing is not performed. Load balancing is a key TE technique in hybrid IP/SDN networks, normally performed by minimizing the maximum link utilization (MLU) through an equal spreading of the traffic load among links (ECMP protocol) [21]. Load balancing allows for a reduction of the OPEX in networks, as well as for a growth of the user base due to the low average link utilization derived from its usage [22, 23]. In hybrid IP/SDN networks, this kind of optimization was introduced in [24]. This proposal separated flows into two categories: *uncontrollable flows*, i.e., flows that were routed by legacy IP equipment, and thus, could not be controlled, and *controllable flows*, which could be controlled because they made use of SDN switches. This work formulated the problem of MLU minimization in hybrid IP/SDN networks by using ECMP techniques in controllable flows. Furthermore, a fully polynomial time approximation scheme (FPTAS) was also proposed. In [25], Ren et al. achieved a near-optimal TE in hybrid IP/SDN networks in polynomial time by proposing a distributed algorithm named TEDR. TEDR relaxed and decomposed the original problem, which was formulated using linear programming, into multiple subproblems related to each of the SDN switches. A small TEDR controller associated with each SDN switch then solved the subproblem assigned to the switch and broadcasted this local solution to the whole network. Finally, local solutions are compared to find a globally near-optimal solution. TEDR achieved a substantial MLU reduction in hybrid IP/SDN networks with at least a 30% SDN deployment rate.

One of the latest related works on the subject is [26]. In this paper, Guo et al. presented HybridFlow, a load balancing solution centered on scalability. HybridFlow made use of the hybrid mode presented in some SDN switches that allowed them to route most flows using OSPF without involving the SDN controller to reduce the overhead. To do so, HybridFlow detected *crucial flows*: flows that carried a high traffic load and that would be directly routed through critical links using OSPF. These crucial links were detected using a *Variation Slope*, a metric also proposed by this work to identify the links carrying the highest set of flows with

higher traffic loads. Crucial flows are then rerouted to perform load balancing using OpenFlow. While HybridFlow enabled near-optimal load balancing with minimal overhead, it also required a fully deployed SDN topology rather than a hybrid IP/SDN topology. Finally, in [27], the NP-hardness of the traffic engineering problem in hybrid IP/SDN networks was proven, and an algorithm named *SDN/OSPF Traffic Engineering* (SOTE) was proposed to solve it. The main purpose of SOTE was to minimize the MLU of a hybrid IP/SDN network by jointly optimizing the OSPF weight setting of the entire network in order to balance outgoing flows at IP nodes, as well as the traffic splitting ratio of flows that aggregated SDN nodes. SOTE was shown to achieve nearly optimal results in realistic topology, even with only a 10% SDN deployment rate. However, none of these works considered the effects of load balancing on energy consumption, as rerouting and the use of the ECMP protocol make use of more links, which in turn requires more links to be powered on, increasing the network's overall energy consumption.

Although both the energy consumption problem and the traffic load balancing problem have been studied separately in hybrid IP/SDN networks, studying them as a joint problem, considering the trade-off between the optimization of routing to maximize energy efficiency and its optimization for MLU minimization, has not been addressed in prior works. Thus, to solve this joint problem and provide a guaranteed QoS, this paper proposes a multi-objective optimization problem, including both its definition and formalization. In particular, two metrics are considered as the minimization objectives: the power consumption of the network and the MLU. This work extends our previous work already published in [28] by performing an evaluation of the impact that SDN nodes have on the flows number and on the amount of traffic that can be controlled, analyzing the impact of the initial weight setting of SDN links on the obtained results and proposing an additional genetic algorithm (GA)-based approach for the power saving phase. Moreover, for this solution to be able to scale, and thus, for it to become practically solvable in large topologies, we also present the Hybrid Spreading Load Algorithm (HSLA), a heuristic solution for this joint problem. Concretely, the main contributions of this work are as follows:

- The consideration of the trade-off between energy consumption and load balancing during the migration from legacy IP to SDN-enabled networks.
- The definition of the joint, multi-objective energy efficiency, and the load balancing problem in hybrid IP/SDN networks that fairly pursues both goals.
- The evaluation of the impact of different SDN selection methods on flows number and on the amount of traffic that can be controlled.
- The evaluation of the impact of the initial weight setting of SDN links on the aforementioned metrics.
- The proposal of the HSLA heuristic algorithm, including a GA-based approach for the link switch off process, which enables the practical and timely solution of the problem in large topologies.
- The comparison of HSLA with existing solutions from both the energy efficiency and the load balancing state-of-the-art within realistic conditions.

The rest of this paper is organized as follows. Section 2 describes and defines the problem. The HSLA algorithm is described in Section 3. Section 4 reports experimental results and their analysis. Ultimately, Section 5 draws some conclusions and future works.

2 Problem definition

We consider a hybrid IP/SDN network modeled as a directed network graph $\mathcal{G} \Leftarrow \mathcal{N} \Leftrightarrow \mathcal{L} \Rightarrow$, where \mathcal{N} is the nodes set and \mathcal{L} is the directed links set. The set of nodes \mathcal{N} is further divided in two subsets: SDN nodes, $\mathcal{N}_{SDN} \in \mathcal{N}$, or IP nodes, $\mathcal{N}_{IP} \in \mathcal{N}$, where $\mathcal{N}_{SDN} \cup \mathcal{N}_{IP} = \mathcal{N}$. With respect to the set of links, each link $l_{ij} \in \mathcal{L}$ connecting node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$ has a capacity of C_{ij} units to accommodate traffic flows, and a power consumption referred to as P_{ij} .

Traffic matrix \mathcal{T} models the traffic and is composed of a set of demands $\delta_{s,d} \in \mathcal{T}$ from each source node $s \in \mathcal{N}$ to each destination node $d \in \mathcal{N}$ in the network. In other words, $\delta_{s,d} \in \mathcal{T}$ refers to the traffic demand entering the network from node s and leaving it through node d .

The goal of the problem is to find an optimal network configuration that jointly minimizes the maximum link utilization and the network power consumption.

Moreover, to define the mixed integer linear programming (MILP) formulation aimed at solving the multi-objective and the multi-commodity-flow (MO-MCF) optimization problem, the following set of variables is required:

- y_{ij} is a binary variable that represents the operational mode of link l_{ij} . Its value is $y_{ij} = 1$ if the link is active. Otherwise, if $y_{ij} = 0$, the link is turned off;
- $f_{ij}^{s,d}$ is an integer variable that represents the amount of traffic demand $\delta_{s,d} \in \mathcal{T}$ that is routed on link l_{ij} ;
- n_{ij}^d is a binary variable indicating whether node i uses node j as the next hop in its forwarding table to reach destination node d ;
- c_i^d is an integer variable representing the cost of the path to go from node i to node d ;

- h_{ij} is an integer variable indicating the interior gateway protocol (IGP) weight of link i, j , which is considered by the IP routers to determine the shortest path routing;
- U is a real variable indicating the maximum link utilization (MLU) in the network.
- \mathcal{N}_i^+ and \mathcal{N}_i^- are subsets of \mathcal{N} that contain the nodes to which node i has *incoming* and *outgoing* links, respectively.

Considering the variables just described, the optimization problem has as goal the jointly minimization of the network power consumption, as well as its maximum link utilization (MLU). For this purpose, two objective functions are needed to solve the multi-objective optimization problem, that are defined in Eqs. (1–2) as follows:

$$F_1 = \min \sum_{l_{ij} \in \mathcal{L}} y_{ij} \cdot P_{ij} \quad (1)$$

$$F_2 = \min U \quad (2)$$

The minimization of the global network power consumption is represented by Eq. 1, while the objective pursued by Eq. 2 minimizes the MLU. Therefore, the MO-MCF problem is defined by the joint optimization of both functions as follows:

$$\min [F_1; F_2] \quad (3)$$

subject to:

$$\sum_{j \in \mathcal{N}_i^+} f_{ij}^{s,d} - \sum_{j \in \mathcal{N}_i^-} f_{ji}^{s,d} = \begin{cases} \delta_{s,d} & \text{if } i = s \\ -\delta_{s,d} & \text{if } i = d \\ 0 & \text{if } i \neq s, d \end{cases} \quad \forall i, s, d \in \mathcal{N}, \delta_{s,d} \in \mathcal{T} \quad (4)$$

$$\sum_{\delta_{s,d} \in \mathcal{T}} f_{ij}^{s,d} \leq y_{ij} \cdot C_{ij} \quad \forall i, j \in \mathcal{L} \quad (5)$$

$$\frac{1}{C_{ij}} \sum_{\delta_{s,d} \in \mathcal{T}} f_{ij}^{s,d} \leq U \quad \forall i, j \in \mathcal{L} \quad (6)$$

$$f_{ij}^{s,d} \leq n_{ij}^d \cdot \delta_{s,d} \quad \forall i, j \in \mathcal{L} : i \in \mathcal{N}_{IP}, \delta_{s,d} \in \mathcal{T} \quad (7)$$

$$\sum_{j \in \mathcal{N}_i^-} n_{ij}^d = 1 \quad \forall i, d \in \mathcal{N} : i \neq d \quad (8)$$

$$0 \leq c_j^d + h_{ij} - c_i^d \leq (1 - n_{ij}^d) \cdot M \quad \forall i, j \in \mathcal{L}, d \in \mathcal{N} \quad (9)$$

$$1 - n_{ij}^d \leq c_j^d + h_{ij} - c_i^d \quad \forall i, j \in \mathcal{L}, d \in \mathcal{N} \quad (10)$$

$$1 \leq h_{ij} \leq h_{\max} \quad \forall i, j \in \mathcal{L} \quad (11)$$

Classical flow conservation constraints are described in Eq. 4, whereas the capacity constraint on links is represented by Eq. 5, where the available capacity upper binds the total traffic routed on a link. Specifically, if the link is in on status ($y_{ij} = 1$), all of its capacity can be employed to transport the traffic; otherwise, its capacity is equal to 0 and the link cannot be used. Equation 6 leads the variable θ to be equal to or greater than the MLU in the network. In the presented model, IP routers are constrained to use a single and shortest path routing and to forward traffic according to the destination. These aspects are involved in the problem formulation through the constraints reported in Eqs. 7–11. Especially, Eq. 7 models destination-based forwarding by allowing the IP router i to send traffic destined to node d over link i, j only if node j is the next hop for the destination ($n_{ij}^d = 1$). Equation 8 imposes the single path routing for IP routers, and specifically it forces node i to choose only one next hop to reach destination node d . It is worth mentioning that, in spite of the fact that SDN nodes can forward traffic over multiple paths, it is important to impose Eq. 8 to all the nodes (not limited only to IP routers). In this way, IP routers are forced to consider only single and shortest paths to forward the traffic, which are calculated according to the weight variables (h_{ij}). Anyway, SDN nodes are able to override these rules since the next hop variables for these types of nodes are not logically bound to the traffic forwarding (Eq. 7 is imposed only on IP routers). Finally, 9–11 enforce the shortest path policy. They impose that if $n_{ij}^d = 1$, then the cost of the path between the nodes i and d must be equal to the weight of the arc connecting it with its neighbor, j , (h_{ij}) including the cost of the path between j and d .

Since the presented problem formulation is NP-hard [29], next, a heuristic approach is presented to solve it in polynomial time.

3 HSLA

The proposed algorithm, named Hybrid Spreading Load Algorithm (HSLA), which aims at solving the MILP problem defined in Section 2, is described in next subsections by pointing out algorithmic aspects and architectural details.

3.1 Exploiting SDN nodes for traffic splitting

The HSLA concept is to take advantage by using the generalized forwarding of the SDN paradigm to split the traffic entering the set of SDN nodes to balance the traffic load. Therefore, an SDN node can split a reaching flow over the rest of the outgoing ports to ensure that the traffic is balanced

toward the next hop. The main benefits of doing this according to the generalized forwarding and the SDN control plane are (i) the traffic can be accurately divided over several outgoing paths and (ii) the considered paths do not need to be the shortest ones. These two points can be summarized according to the concept of multipath routing (MPR). In contrast, traditional IP routers, which work according to destination-based forwarding and rely on shortest path routing protocols (e.g., OSPF [10]), can split traffic over equal cost multipaths (ECMPs, [21]), but this operation has several limitations. Firstly, traffic splitting is based on flow hashing methods that allow to balance only in terms of flow number (not considering the actual traffic volume) and to lead to low accuracy. Furthermore, only the shortest paths can be considered, thus, reducing the load balancing opportunities. Finally, all these decisions are delegated to the local control plane agent with logic that cannot be centrally controlled to achieve a global objective. For this reason, in this paper, we decide not to rely on the ECMP mechanism of the IP routers, allowing traffic splitting only at SDN nodes, where this process is effective and flexible as a result of the availability of the routing mechanism of group tables or using specific flow rules. In particular, we assume that the capabilities of SDN switches are exploited to perform accurate traffic flow over multiple shortest paths.

The idea behind the proposed routing scheme is represented through Figs. 1, 2, and 3, where an 8-node hybrid IP/SDN network is reported. We are interested to the path that is evaluated to route the flow traffic from node A to node H. In Fig. 1, the resulting path for the case where all the nodes are legacy IP nodes ($P_{A,H} = \{A - C - F - H\}$) is represented by the blue line. For the sake of simplicity, all the weights for links are here set to 1. In this case, destination-based forwarding is applied, such as in traditional routing protocols, e.g., OSPF [10].

Figure 2 represents the scenario where a traditional IP node is replaced by an SDN node at a particular stage in the transition from IP to SDN and the selection method for such replacement indicates that the upgraded node must be C. In this case, the links associated with the SDN node

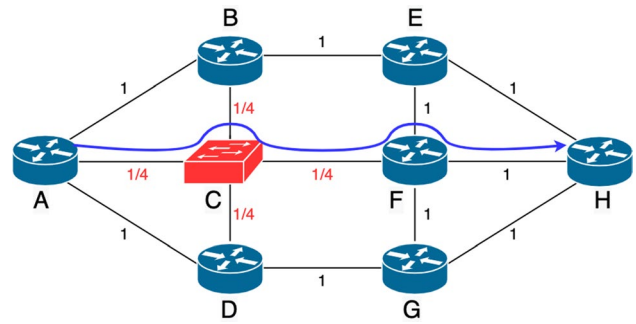


Fig. 2 A to H path in a hybrid IP/SDN network considering C as SDN node

are weighted by decreasing their values from $h_{i,j} = 1$ (initial setting) to $h_{i,j} = \frac{h_{i,j}^-}{k_i} = \frac{1}{4}$, where the weight of link $l_{i,j}$ before the update is represented by $h_{i,j}^-$ and k_i is a metric that purposes to attract the flows to node i , e.g., its degree. The decreasing of the link weights related to SDN nodes is done to force the flows to pass through them, leading to balance the traffic among different links exploiting the ECMP protocol [21]. Despite this, in the reported example, although link weights are modified, the single SDN node replacement does not affect the path followed by flow A – H (blue line) with respect to the case of Fig. 1.

By upgrading a second node to SDN, e.g., G, the flow arriving at node C is equally divided among the set of shortest paths from it toward the destination (see Fig. 3). In the example, two shortest paths with the same cost exist from C to H: $P_{C,H}^1 = \{C - D - G - H\}$ and $P_{C,H}^2 = \{C - F - G - H\}$, which are represented by green lines. Thus, the traffic volume represented by the blue line is split by two and steered through the aforementioned (green) paths. With this traffic division, the proposed solution targets the traffic load to be balanced and the MLU to be reduced through the use of the ICMP protocol at the SDN nodes.

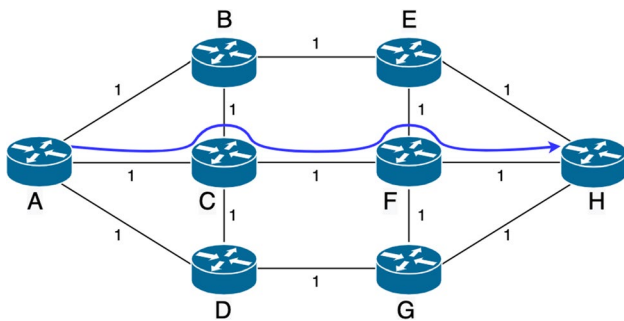


Fig. 1 A to H path in a legacy IP network

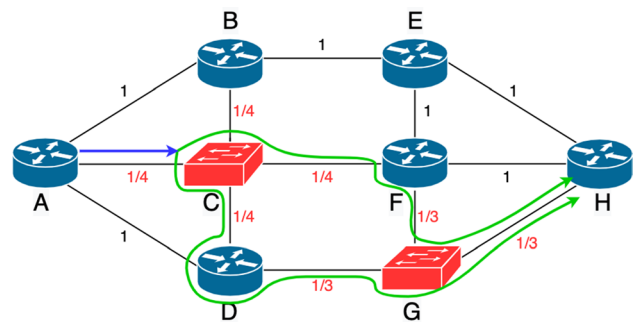


Fig. 3 A to H path in a hybrid IP/SDN network considering C and G as SDN nodes

3.2 HSLA description

Next, we present the proposed solution, which is composed of 2 algorithms. The former aims at spreading the traffic through the subset of SDN nodes as a result of using the ECMP protocol (pseudo-code reported in Algorithm 1). Once the traffic is

handled, the second algorithm tries to save energy by switching off links. In addition, for the link switch off (LSO) process, two solutions have been considered: the Less Loaded Links Algorithm (L3A), which iteratively removes the less loaded link in the network (see Algorithm 2) and repeats and the Genetic Algorithm-based LSO (GA-LSO), which

Algorithm 1 Pseudo code of the proposed Hybrid Spreading Load Algorithm (HSLA).

Require: A network graph: $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, a traffic matrix: \mathcal{T}

```

1: create  $\mathcal{W} \leftarrow \emptyset$ 
2: create  $\mathcal{R} \leftarrow \emptyset$ 
3: for all  $l_{i,j} \in \mathcal{L}$  do
4:   if  $i \in \mathcal{N}_{\text{SDN}}$  then
5:      $w_{i,j} \leftarrow \frac{1}{k_i}$ 
6:   else
7:      $w_{i,j} \leftarrow 1$ 
8:   end if
9: end for
10: for all  $\delta_{s,d} \in \mathcal{T}$  do
11:   create  $p_{s,d} \leftarrow \{\}$ 
12:   create  $\lambda \leftarrow \text{false}$ 
13:    $P \leftarrow k\text{-shortest\_paths}(\delta_{s,d}, \mathcal{G})$ 
14:   while  $\lambda == \text{false}$  do
15:     if  $p \in P$  is an IP path then
16:        $b \leftarrow \text{check\_path\_constraints}(\delta_{s,d}, p, \mathcal{G})$ 
17:       if  $b == \text{true}$  then
18:          $\mathcal{R} \leftarrow \text{assign\_flow\_to\_path}(\delta_{s,d}, p, \mathcal{G})$ 
19:          $p_{s,d}.\text{concatPath}(p)$ 
20:          $\lambda \leftarrow \text{true}$ 
21:       end if
22:     else
23:       for all  $l_{i,j} \in p$  do
24:         if  $i \in \mathcal{N}_{\text{SDN}}$  then
25:            $P^* \leftarrow k\text{-shortest\_paths}(\delta_{i,d}, \mathcal{G})$ 
26:            $u \leftarrow \text{num\_paths\_same\_cost}(P^*)$ 
27:           for all  $p' \in P^*$  with same cost do
28:              $b \leftarrow \text{check\_path\_constraints}(\frac{\delta_{s,d}}{u}, p', \mathcal{G})$ 
29:             if  $b == \text{true}$  then
30:                $\mathcal{R} \leftarrow \text{assign\_flow\_to\_path}(\frac{\delta_{s,d}}{u}, p', \mathcal{G})$ 
31:                $p_{s,d}.\text{concatPath}(p')$ 
32:             else
33:               clear incomplete flow assignments
34:               check next path  $p \in P$  (line 14)
35:             end if
36:           end for
37:            $\lambda \leftarrow \text{true}$ 
38:         else
39:            $b \leftarrow \text{check\_link\_constraints}(\delta_{s,d}, l_{i,j}, \mathcal{G})$ 
40:           if  $b == \text{true}$  then
41:              $\mathcal{R} \leftarrow \text{assign\_flow\_to\_link}(\delta_{s,d}, l_{i,j}, \mathcal{G})$ 
42:              $p_{s,d}.\text{addLink}(l_{i,j})$ 
43:           else
44:             clear incomplete flow assignments
45:             check next path  $p \in P$  (line 14)
46:           end if
47:         end if
48:       end for
49:     end if
50:   end while
51: end for
52: return feasible routing matrix  $\mathcal{R}$  if it exists

```

▷ Set for link weights
 ▷ Routing matrix
 ▷ Update link weights
 ▷ Find path $\forall \delta_{s,d} \in \mathcal{T}$
 ▷ Path for demand $\delta_{s,d}$
 ▷ Boolean for path found
 ▷ Check each path $p \in P$
 ▷ No SDN nodes
 ▷ Check link constraints and no loops
 ▷ There are SDN nodes in the path
 ▷ It is an SDN node
 ▷ It is an IP node

exploits the use of GAs to power off links and steer the traffic among the remaining subset of links (Algorithm 3).

Starting with the HSLA, whose goal is the minimization of the MLU, 2 parameters are required as input. The first parameter is the network graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, with $\mathcal{N}_{TP} \in \mathcal{N}$ as the set of legacy IP nodes, $\mathcal{N}_{SDN} \in \mathcal{N}$ composed of the set of SDN nodes, and $\mathcal{N}_{SDN} \cup \mathcal{N}_{IP} = \mathcal{N}$. The second parameter is the traffic matrix \mathcal{T} to be satisfied. First, an initialization phase for setting the link weights is performed in lines (1 – 9). As introduced in Section 3.1, $h_{ij} = 1$ is set for the weight for links connecting two IP nodes, whereas they are decreased to $h_{ij} = \frac{1}{k_i}$ when they connect an SDN node i , with k_i as the related metric representing the node importance.

Once the initial weight setting is performed, the objective of the HSLA is to provide for all the traffic demands contained in the traffic matrix \mathcal{T} passed as input, i.e., to find a feasible routing matrix \mathcal{R} for \mathcal{T} that minimizes the MLU. For this aim, each traffic demand (s, d) is iteratively handled. First, the k shortest paths are evaluated from source node s to destination d according to destination-based forwarding (line 13).

Then, each path in the reported set is verified to check if it is a *noncontrollable path* (all the nodes are IPs) or a *controllable path* (at least one node is an SDN). In the former case (lines 15 – 23), a classic destination-based forwarding is employed to steer the traffic, checking that link constraints are satisfied and no loops are formed (line 16). If the SDN nodes are found in the evaluated path (it is a *controllable path*), the heuristic verifies each node in the path from s to d to discern whether they are SDN or IP nodes (lines 24 – 49). If the checked node i is an SDN, the k shortest path procedure is executed again (line 27) to get the subset of u paths with the same cost from the SDN node i to the destination d . Then, the traffic demand (s, d) arriving at i is equally divided among the subset of u paths with an equal cost to d and a volume of $\frac{(s, d)}{u}$ (lines 29 – 37). For each evaluated

path, link constraints and the presence of loops are also checked (line 28). If the evaluated node i is an IP, the traffic demand is fully routed over the outgoing link without performing traffic splitting (lines 39 – 47). This process is repeated until all the nodes in the path from s to d are evaluated and the traffic is successfully steered, satisfying link constraints throughout of the path and the avoidance of loops. Finally, the output reports a feasible routing matrix \mathcal{R} .

3.3 Less Loaded Links Algorithm

Once Algorithm 1 evaluates a routing solution where the traffic load is balanced and thus the MLU is minimized, a feasible network configuration that minimizes the required power consumption is found by a second algorithm. Algorithm 2 reports the pseudo code of the L3A. It takes as input the routing matrix outputted by the HSLA, \mathcal{R} , as well as the network graph and the traffic matrix to be served. Initially, the set of links are sorted according to their load in ascending order (line 4). Then, each link is removed (turned off) from the network topology, and the HSLA is executed again to find feasible routing (if possible). If there is no feasible routing, the link is returned to the topology (switched on), and the next link is considered. This process is repeated until all the network links are evaluated. Ultimately, a feasible routing matrix, \mathcal{R}' , is reported, which (i) balances the link load and (ii) reduces the network power consumption.

3.4 Genetic algorithm-based link switch off

The second algorithm that is proposed for the power saving procedure is based on GAs, a type of heuristic based on the mechanisms of natural evolution where individuals reproduce and the more skilled ones survive [30].

To solve the problem, a chromosome representing a solution must be defined. In our case, a chromosome c represents a network configuration in terms of link activity (Eq. 12). In

Algorithm 2 Pseudo code of the proposed Less Loaded Links Algorithm (L3A).

Require: $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, \mathcal{T} , the routing matrix obtained by Alg. 1: \mathcal{R}

- 1: create $\mathcal{G}' \leftarrow \emptyset$ ▷ Aux network graph
- 2: create $\mathcal{L}^* \leftarrow \emptyset$, $\mathcal{L}' \leftarrow \mathcal{L}$ ▷ Aux links set
- 3: create $\mathcal{R}' \leftarrow \emptyset$ ▷ Output routing matrix
- 4: $\mathcal{L}^* \leftarrow \text{sort_links_by_load}(\mathcal{G}, \mathcal{R}, \text{'ascend'})$
- 5: **for all** $l_{i,j} \in \mathcal{L}^*$ **do**
- 6: $\mathcal{L}' \leftarrow \mathcal{L}' - l_{i,j}$ ▷ Remove link from the set
- 7: $\mathcal{G}' = (\mathcal{N}, \mathcal{L}')$
- 8: $\mathcal{R}' \leftarrow \text{HSLA}(\mathcal{G}', \mathcal{T})$
- 9: **if** \mathcal{R}' is not feasible **then**
- 10: $\mathcal{L}' \leftarrow \mathcal{L}' \cup l_{i,j}$
- 11: **end if**
- 12: **end for**
- 13: **return** feasible routing matrix \mathcal{R}' if it exists

particular, each gene, g_k , of chromosome c is a binary value representing the operational mode of the k -th link. If link k is powered on, then $g_k = 1$. Otherwise, if the link is powered off, then $g_k = 0$. This representation fits with the philosophy of canonic GAs, which use binary strings to represent individuals.

$$c = \{g_1, g_2, \dots, g_L\}, g_k \in \{0;1\} \quad (12)$$

Each potential solution represented by a chromosome must be evaluated to obtain its suitability to the problem, i.e., its goodness, according to a fitness function. In this problem, the fitness function must check if the network configuration mapped by the chromosome ensures that the routing constraints (link capacities) and whether all the traffic demands are satisfied. If the feasibility check is correct, then the resulting fitness value is the number of powered links. The goal of the GA-LSO is, therefore, to minimize this number to ensure that the power savings are increased.

In the following, the description of the proposed GA-based LSO algorithm is provided, including the operators and functions used to carry out the evolutionary process. It takes six parameters as input: (i) the network graph, $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; (ii) the routing matrix obtained by HLSA, \mathcal{R} ; (iii) a traffic matrix, \mathcal{T} ; (iv) the population size, κ ; (v) the mutation rate, m_r ; and (vi) the maximum generations number that compose the evolutionary process, θ .

At the beginning, four variables are initialized: gen as an indicator of the current generation, P_{gen} as the population

of individuals to be evaluated (initially empty), and the two variables that will be returned as output: sol_{best} as the best solution found and $fval_{best}$ as the numerical goodness of such a solution. The latter variable is initialized to a large value to be minimized during the execution of the GA-based LSO.

In the first step, the GA creates the initial population, P_{gen} , of size κ with the set of individuals to be evaluated by the fitness function. Next, each individual $c \in P_{gen}$ is evaluated by applying the fitness function previously described, and the best solution found is iteratively updated (lines 3 – 9). The evolutionary process starts at line 10 with the selection of the individuals that will be part of the next generation. In particular, *crossover* and *mutation* processes are applied to the current population, P_{gen} , to create offspring (*children*) that will be evaluated in the following generations. Specifically, single-point crossover and uniform mutation are applied. The mutated *children** obtained after applying the genetic operations are evaluated by the fitness function, and the best solution found is again iteratively updated (lines 15 – 21). The process ends after the execution of θ iterations, i.e., the maximum generations number passed as input. The algorithm returns the best solution found, sol_{best} , and its associated fitness value, $fval_{best}$. The best solution represents the best network configuration in terms of power savings (least number of active links), which respects the routing and capacity constraints.

Algorithm 3 Pseudo code of the proposed GA-based LSO algorithm.

Require: A network graph: $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, the routing matrix obtained by Alg. 1: \mathcal{R} , a traffic matrix: \mathcal{T} , population size: κ , mutation rate: m_r , maximum generations: θ

```

1: create  $gen \leftarrow 0, P_{gen} \leftarrow \emptyset, sol_{best} \leftarrow \emptyset, fval_{best} \gg L$ 
2:  $P_{gen} \leftarrow createPopulation(\mathcal{G}, \kappa, uniform)$ 
3: for all chromosome  $c$  in  $P_{gen}$  do
4:    $(fval_c, sol_c) \leftarrow fitness(c, \mathcal{G}, \mathcal{R}, \mathcal{T})$ 
5:   if  $fval_c < fval_{best}$  then
6:      $sol_{best} \leftarrow sol_c$ 
7:      $fval_{best} \leftarrow fval_c$ 
8:   end if
9: end for
10: while  $gen \leq \theta$  do
11:    $parents \leftarrow selectParents(P_{gen}, roulette)$ 
12:    $P^* \leftarrow P_{gen} \setminus parents$ 
13:    $children \leftarrow performCrossover(parents, single-point)$ 
14:    $children^* \leftarrow performMutation(children, m_r, uniform)$ 
15:   for all chromosome  $c$  in  $children^*$  do
16:      $(fval_c, sol_c) \leftarrow fitness(c, \mathcal{G}, \mathcal{R}, \mathcal{T})$ 
17:     if  $fval_c < fval_{best}$  then
18:        $sol_{best} \leftarrow sol_c$ 
19:        $fval_{best} \leftarrow fval_c$ 
20:     end if
21:   end for
22:    $P^* \leftarrow P^* \cup children^*$ 
23:    $gen \leftarrow gen + 1$ 
24:    $P_{gen} \leftarrow P^*$ 
25: end while
26: return  $sol_{best}, fval_{best}$ 

```

4 Experimental results

In this section, we provide an experimental evaluation of the proposed solution. Firstly, the simulation environment is reported. Next, an analysis is provided to evaluate different methods for the selection of the nodes to be migrated from IP to SDN, both in terms of controllable flows and the amount of controllable traffic. We adopt the definition given in [24] to refer to *controllable flows*. A *controllable flow* is a flow that is routed by SDN equipment. In contrast, an *uncontrollable flow* is a flow that is completely routed by legacy IP equipment, and thus, cannot be controlled. Therefore, controllable traffic is the amount of traffic associated with the set of controllable flows. Moreover, the impact of the weight setting of links connecting SDN nodes is also evaluated by considering different approaches. To analyze the benefits of applying the HSLA over topologies of different sizes and traffic loads, a thorough performance analysis is carried out. Metrics, such as MLU and power savings, are evaluated. Finally, a comparison with two representative state-of-the-art solutions that separately address the load balancing and the energy consumption problem in hybrid IP/SDN networks is also performed.

4.1 Simulation setup

In the simulations, we consider three network topologies of different sizes: Nobel (17 nodes, 52 links), Geant (22 nodes, 72 links), and Germany (50 nodes, 176 links). To evaluate the variability of the daily traffic patterns in each network, a set of traffic matrices are retrieved from [31]. In particular, 5 TMs are considered for the analyses by scaling down the peak TM in the data set by a factor of 0.4, 0.5, 0.7, and 0.9, respectively.

Link capacities are set as follows. First, we select the peak TM, and we route it over a set of shortest paths derived after applying the Dijkstra algorithm on the network graph. After this step, each link $l_{i,j}$ carries an amount of traffic $t_{i,j}$. Then, we assume that the capacity of each link can be upgraded by installing a set of line cards. A line card has a capacity of Δ_C equal to $0.5 \max_{i,j} t_{i,j}$, i.e., half of the traffic carried by the link with the highest link utilization. Finally, we consider installing the minimum line cards number that is needed by each link to make their utilization less than 100%.

In our analysis, we assume that all the network links have the same power consumption when they are active ($P_{i,j} = 1$). The transition from the IP to SDN networks is performed step-by-step, i.e., by upgrading one IP legacy node to SDN at a time. At each stage, one node is selected as the most suitable to be upgraded to the SDN. For the selection process, four methods are considered: (i) highest degree first (HDF),

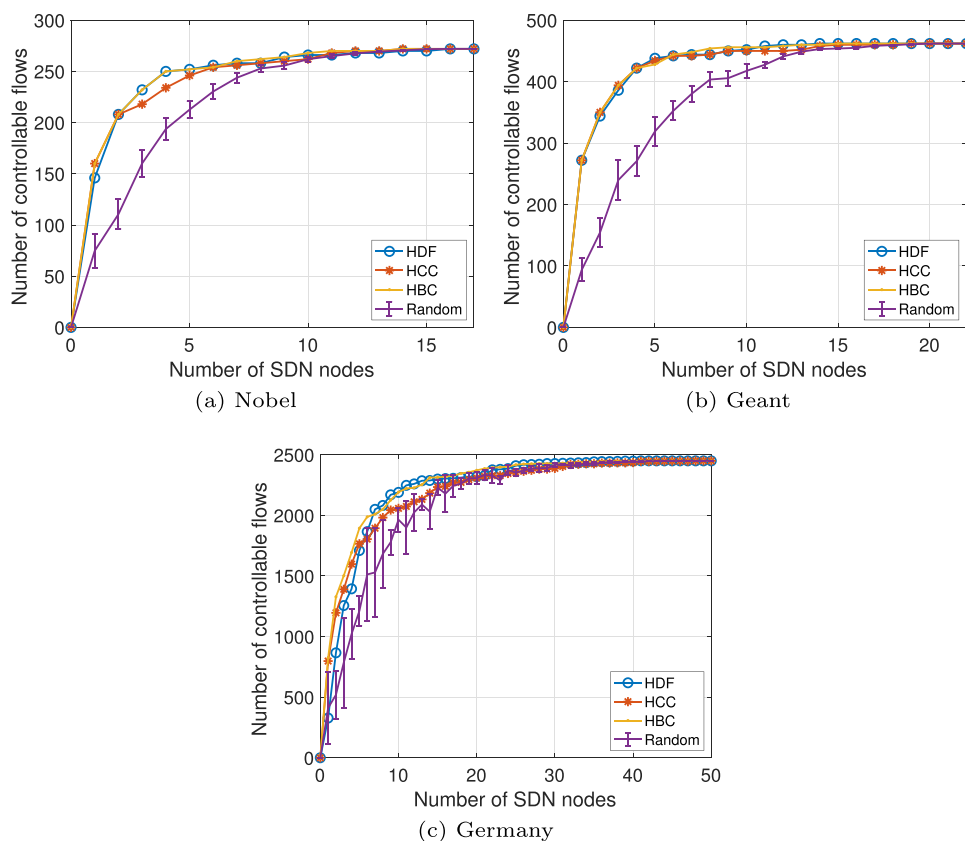
which sorts the set of potential nodes (remaining IP nodes) according to their degree, i.e., the number of incoming and outgoing links; (ii) highest closeness centrality (HCC), to select the node having the shortest distance to the rest of the nodes; (iii) highest betweenness centrality (HBC), with the aim of obtaining the node that is involved in most of the shortest paths among the nodes in the network; and (iv) random.

The motivation for the choice of the aforementioned selection methods is explained next. If the upgrade of IP nodes to the SDN is performed according to the HDF method, it means that, after each stage, the controller will have the largest possible amount of information under control compared with the case where other potential nodes would be selected. Thus, the node with the highest degree is chosen so that the controller will be able to control the highest number of ports through, e.g., the use of the *OFPMPP_PORT_STATS* messages. If HCC is the method for the SDN nodes selection, the focus is placed on the distance of a node to the rest of the nodes. This situation is interesting for detecting nodes that are able to spread information very efficiently throughout the network. If a node is selected as an SDN using HCC, it means that the destination node (as well as the source node) is close; thus, the path is shorter with regard to other potential nodes, and improvements in the energy efficiency can be experienced. Ultimately, we evaluate the importance of a node with HBC. In this case, the node with HBC is involved in most of the shortest paths for each source-destination pair. Since the HSLA is based on the computation of the shortest paths both from the source node and from each SDN node (if any) to the destination, the HBC seems to be an interesting option to be exploited in the SDN node selection process. Finally, randomness is also considered to show the benefits of using methods that are based on the features of the network graph.

4.2 Evaluation of the SDN nodes selection method

The first analysis that is proposed aims at evaluating the controllable flows number and the amount of controllable traffic that results after the migration of legacy IP nodes to SDN ones. In particular, Fig. 4 reports the controllable flows number as a function of the SDN nodes number for the Nobel, Geant, and Germany topologies. Four different methods for the selection of the SDN nodes are considered: HDF, HCC, HBC, and random. Simulations were run 25 times, and confidence intervals are shown in the figure. Since the initial link weight setting is important for evaluating the effectiveness of the proposal, the link weight setting of the SDN nodes in this evaluation is set according to their degree, $w_{i,j} = \frac{1}{k_i}$, with i as the SDN node and k_i as its degree. In the case of links connecting only IP nodes, their weight is set to

Fig. 4 Number of controllable flows as a function of the SDN nodes number for Nobel (a), Geant (b), and Germany (c). The link weight setting for the SDN nodes is set according to their degree, $w_{i,j} = \frac{1}{k_i}$



$w_{i,j} = 1$. Looking at Fig. 4, it can be seen that the three methods that are based on network features (HDF, HCC, and HBC) present similar results for small networks (Nobel and Geant), overcoming the one exploiting randomness. In the case of the Germany network (see Fig. 4c), larger differences can be found within the range $[0, 20]$ of SDN nodes: (i) lines for HCC and HBC are close for a small SDN nodes number $[1 - 5]$ with a slightly better performance with regard to HDF; and (ii) the performance of HCC decreases in the range $[5 - 20]$ compared to HDF and HBC, which maintains the best results for all the problem instances. Thus, in general, the HBC selection method is the one that presents the best performance in terms of controllable flows during the transition from IP to SDN.

To verify the previous outcomes, Fig. 5 reports the amount of controllable traffic as a function of the SDN nodes number for the four selection methods and the three topologies. In this case, it can be remarked that HDF struggles with respect to HCC and HBC for a small SDN node number, especially in the case of large networks (see Fig. 5c), where its performance is similar to that of the worst selection method, i.e., random. Thus, from these analyses, we can state that HBC is the selection method that is able to control the highest number of flows and the largest amount of traffic when migrating an IP network to an SDN network.

4.3 Evaluation of the link weight setting method

In this section, a discussion of the importance of selecting an appropriate link weight setting is provided. In particular, we focus on tuning the weights of the links connecting the SDN nodes. For the remaining links uniquely connecting legacy IP nodes, their weight is set to 1, as in the OSPF protocol. This choice derives from the control a network admin is able to carry out on the traffic traversing SDN nodes. As a result, if the definition of the SDN link weights is set according to metrics related to the network structure and the flows number (and traffic) to be handled by the SDN nodes is increased, our proposed algorithm handles a thinner granularity on the traffic.

Three methods for setting the link weights are evaluated: (i) LWS: the degree where the weights of the SDN links are inversely set according to their degree (as in the evaluations of Section 4.2 and the description of Fig. 3); (ii) LWS: the closeness where the node closeness centrality is considered for the link weights; and (iii) LWS: the betweenness where the metric used to assess the link weights is the betweenness centrality of the nodes. In all the cases, links attached to an SDN node are assigned a weight that is inversely proportional to the considered metric (degree, closeness or betweenness).

Fig. 5 Amount of controllable traffic as a function of the SDN nodes number for Nobel (a), Geant (b), and Germany (c). The link weight setting for SDN nodes is set according to their degree, $w_{i,j} = \frac{1}{k_i}$

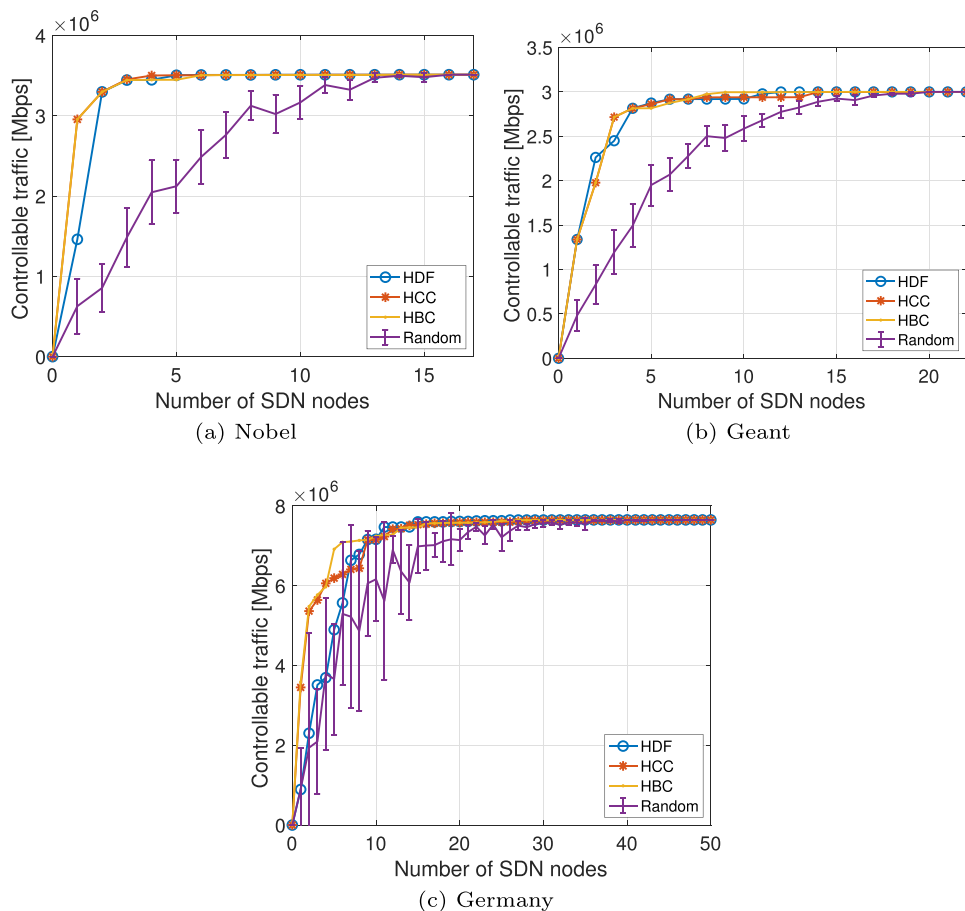


Figure 6 shows the number of controllable flows as a function of the SDN nodes number for Nobel, Geant, and Germany considering the three methods described above. Moreover, the case where all the links in the network have the same weight, i.e., $w_{i,j} = 1$ regardless of whether they are IP or SDN links, is also reported with the legend “No LWS.” In the figures, it can be seen that the link weight setting based on either node degree or node betweenness centrality represents the best choice to increase the number of controllable flows for all the network topologies. The worst-case scenario is reported for the closeness centrality method, which is even worse than the case where no differences are adopted for SDN links and regular IP links, i.e., when no considerations for the link weights are assumed.

These results can be confirmed when looking at Fig. 7, where the amount of controllable traffic is shown as a function of the SDN nodes number. In this case, although the benefits of using a method to set the weights of the SDN links are slightly reduced compared to the number of controllable flows (especially for small networks, see Fig. 7a and b), the metrics based on the node degree and the betweenness centrality clearly outperform the rest. In conclusion, we can state that the impact of selecting an appropriate method for the link weight setting is very important for the

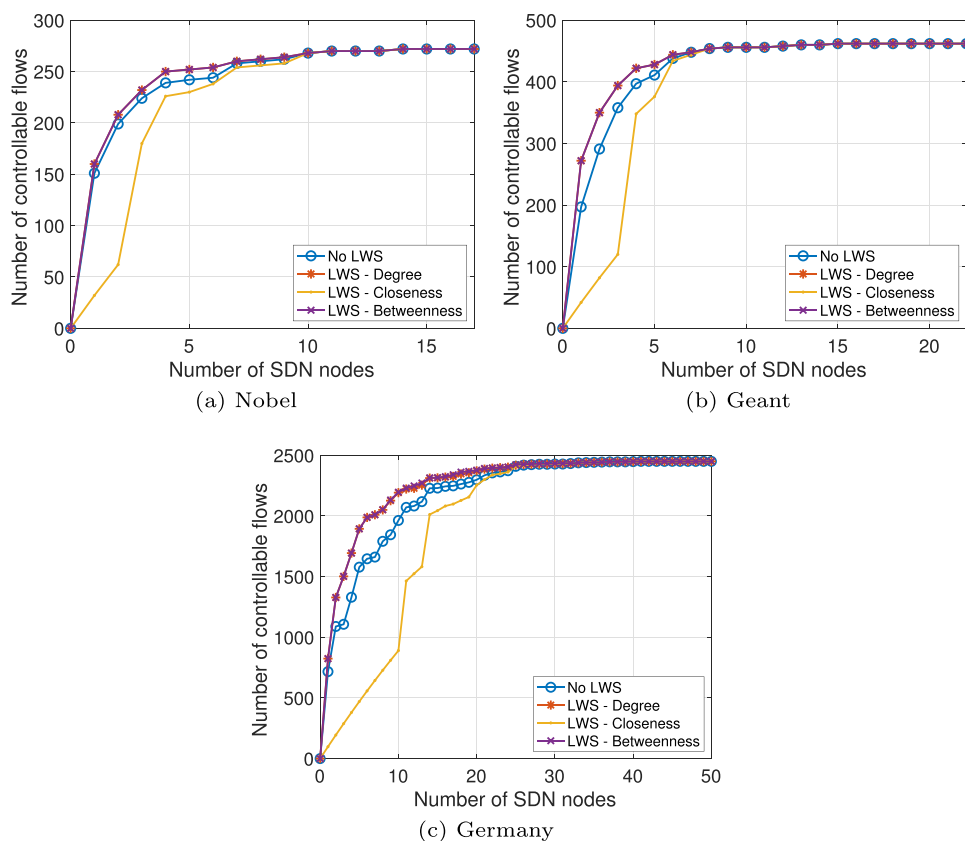
controllability of the flows, and hence, the amount of traffic that can be managed in the hybrid IP/SDN network. Thus, for the evaluation of the proposed solution, we consider the link weight setting based on the SDN node degree.

4.4 HSLA performance evaluation

The goal of the first analysis is to evaluate the effectiveness of the proposed solution considering the joint optimization of energy efficiency and load balancing during the transition from IP to SDN nodes. Figure 8 shows the MLU obtained by the HSLA as a function of the SDN deployment rate for the Nobel topology. Specifically, the results for each of the 5 TMs are accounted for the three SDN node selection methods (HDF, HCC, and HBC). Note that TM 5 is the peak TM in the data set and TM 1 is the non-peak TM.

Figure 8 shows a general trend. The MLU decreases with the SDN deployment rate for a small percentage of migrated SDN nodes. This MLU reduction is different depending on the considered selection method, with an early impact for HDF and HCC and a more progressive impact for HBC. After that, the MLU is stabilized, and adding SDN nodes does not impact the resulting MLU. Clearly, there is an existing direct proportional relationship between the overall

Fig. 6 Number of controllable flows as a function of the SDN nodes number for Nobel (a), Geant (b), and Germany (c). The SDN node selection method is HBC



traffic load that is injected into the network and the obtained MLU. TM 5 represents the worst-case scenario and TM 1 represents the case for non-peak traffic, where more energy saving opportunities can be found.

A considerable decrease in the obtained MLU is tested every time an IP node is replaced by an SDN node when the SDN deployment rate is below 40% both for HDF (Fig. 8a) and HCC (Fig. 8b). After such a percentage, the achieved MLU is approximately constant up to reaching the full SDN scenario (deployment rate of 100%). In other words, the highest impact during the transition from IP to SDN in terms of MLU is found in the range (0, 40)% of the deployment rate. It is in Fig. 8c where the MLU behavior is step-wise. In this case, the benefits of adding a new SDN node are not as high as in the case of HDF or HCC, regardless of the transition stage.

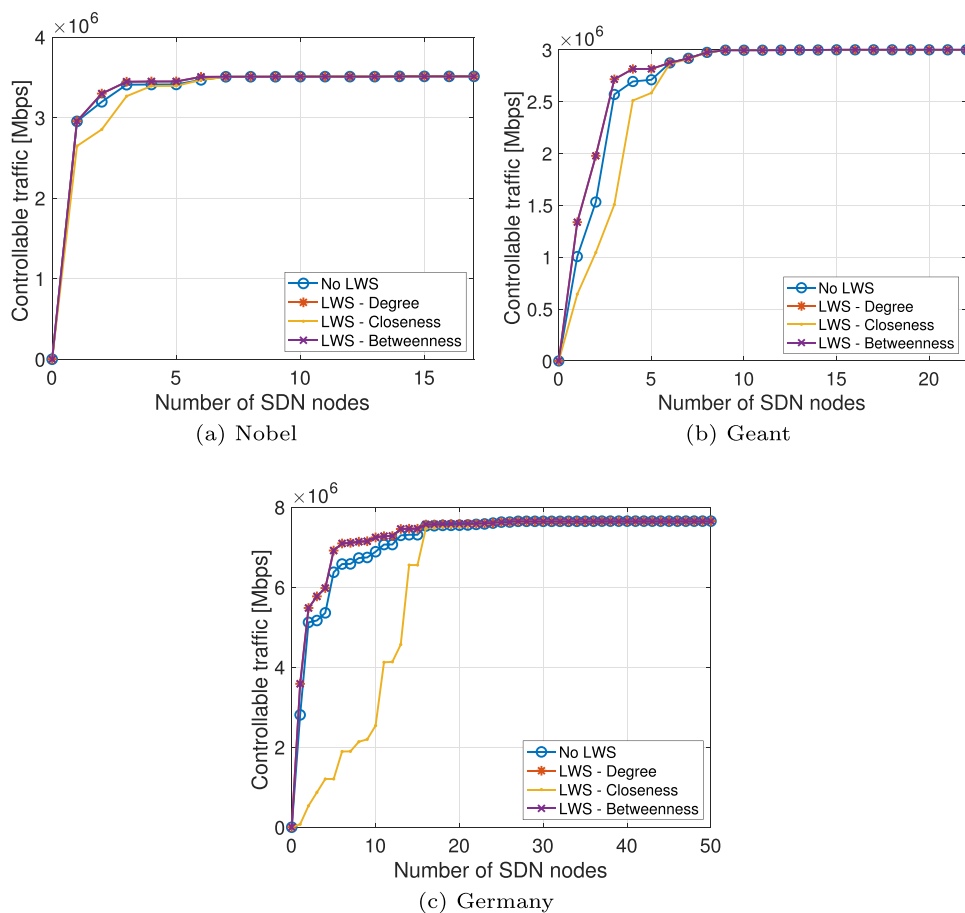
Once the evaluation of the obtained MLU has been carried out, we move our attention to the power savings achieved by the HSLA throughout of the IP/SDN transition process. In Fig. 9, the power savings obtained by L3A are reported for the 5 TMs and the three selection methods, considering different percentages of the SDN deployment rate. Considering the three subfigures, HDF is clearly the method that outperforms the other ones, reaching a power saving peak of approximately 50% when half of the network nodes are upgraded to SDN. In this case, the obtained trend is different

from the one experienced when evaluating the MLU in Fig. 8: the power savings are increased with the SDN deployment rate up to reaching a threshold that slightly exceeds the half of the nodes considered as SDN. After such a value, the power savings sharply decrease, and no substantial differences can be found among traffic matrices and with the increase in the deployment rate. Although the HCC method slightly improves the power savings outcomes with regard to HDF for a deployment rate of 30% and low traffic scenarios (see Fig. 9b), it struggles when the traffic load is increased. Finally, HBC has been found to be the worst selection method for the power savings objective. In summary, the SDN selection method that presents the best results, both for the minimization of the MLU and the minimization of the power consumption, is HDF. An explanation for this outcome is derived by the fact that such a method aims at selecting the node with the highest degree, hence involving the maximum number of links and controllable paths.

4.5 Performance analysis over larger networks

The aim of the next analysis is to evaluate the results of the proposed solution over topologies with different sizes. Specifically, in Fig. 10, the MLU is reported as a function of the traffic scaling factor for the three topologies under test: Nobel, Geant, and Germany50. For each topology, 4

Fig. 7 Amount of controllable traffic as a function of the SDN node number for Nobel (a), Geant (b), and Germany (c). The SDN node selection method is HBC



deployment rate percentages (α) have been considered, approximating 25%, 50%, 75%, and 100% of the nodes in the network. HDF is adopted as the node selection method. In the figures, it can be seen that the MLU increases with traffic, as previously discussed when analyzing Fig. 8 for the three topologies. It is worth to note that the best outcomes for the medium and large topologies are not obtained in the full SDN scenario ($\alpha = 100\%$) except for when half of the nodes are replaced (see Fig. 10b and c). In the Nobel case, the minimal MLU is experienced when $\alpha = 75\%$. Regardless, the difference with the half hybrid IP/SDN network is negligible.

Figure 11 reports power savings outcomes as a function of the traffic load for the three considered topologies and the different values of α . By analyzing Fig. 11, there are differences between the results obtained for small and large topologies. Indeed, in the latter case (Geant and Germany50), a decreasing behavior in the power savings can be taken out with the increase in the traffic scaling factor. Therefore, a higher value of α is generally associated with power saving gains, especially for Geant. It is in the Nobel case (Fig. 11a) where the behavior is the opposite. The best results of power savings are obtained when $\alpha \leq 50\%$, as previously discussed when analyzing Fig. 9a. Once this threshold is overtaken, the power savings are considerably decreased, regardless of the network traffic load.

The explanation for the obtained power saving results is as follows. The proposed algorithm aims at either balancing the traffic load or reducing the network power consumption. In the first stage, for the HSLA, the link weights are adjusted to create a large portion of the controllable traffic. As a consequence, the main role of the SDN nodes is to balance the traffic and not to reduce the power consumption (aim of the second stage, L3A). At the same time, the weight selection logic becomes less effective when the deployment ratio increases (approaching a full SDN network) since the amount of controllable traffic is already high. Thus, the proposed algorithm accomplishes its performance peak when the network is half IP-half SDN.

In the following, the other power saving approach that has been defined, namely, GA-LSO, is analyzed. Figure 12 shows the power savings achieved by GA-LSO as a function of the traffic scaling factor for the three considered topologies. In general, GA-LSO obtains better results than L3A, especially for small networks (see Figs. 11a and 12a for Nobel). The expected decreasing tendency of the power savings with the increase in the traffic load is confirmed for all cases, and the average power saving gain is 4.7% for a 50 – 50% hybrid IP/SDN network ($\alpha = 0.5$), increasing up to 7.1% in the case of a full SDN migration ($\alpha = 1$). In the

Fig. 8 MLU value with respect to the percentage of SDN nodes for Nobel topology considering HDF (a), HCC (b) and HBC (c) selection methods

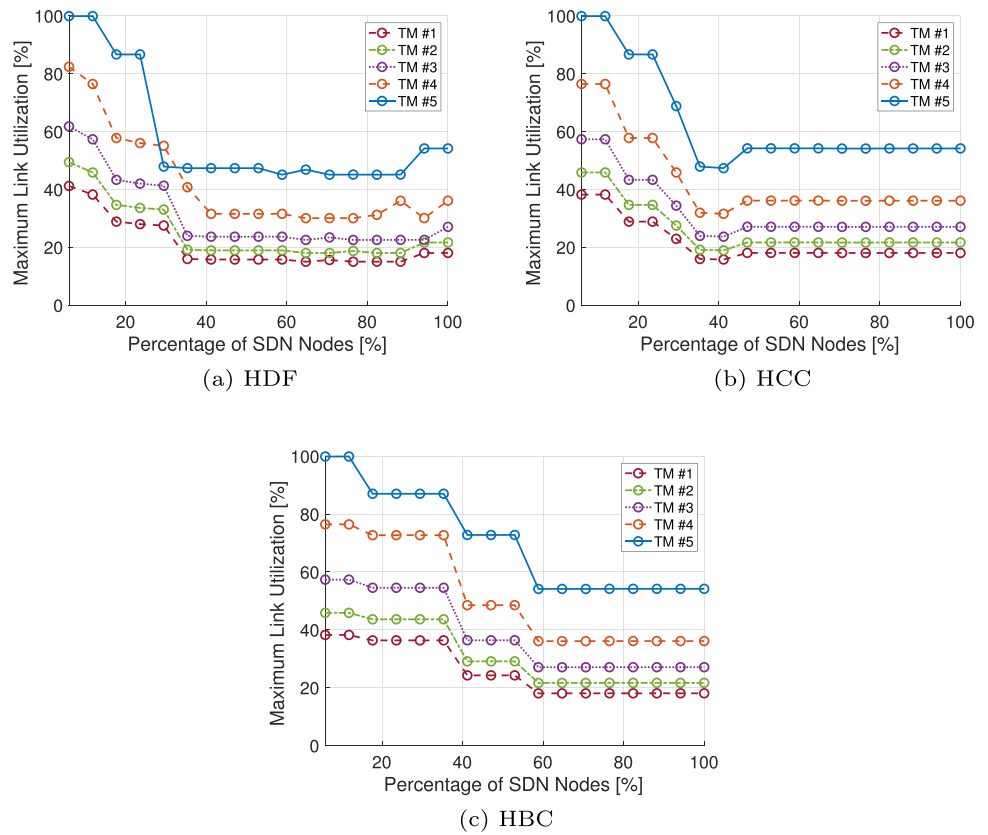


Fig. 9 Percentage of power savings obtained by L3A with respect to the percentage of SDN nodes for Nobel topology considering HDF (a), HCC (b) and HBC (c) selection methods

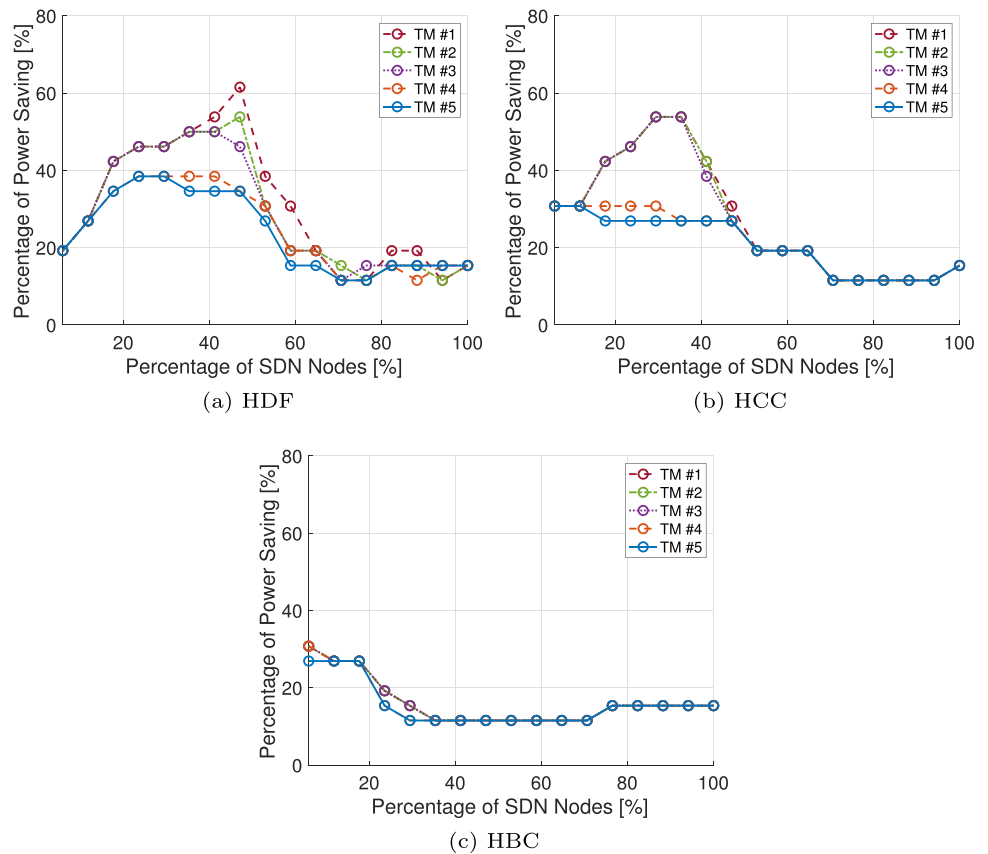


Fig. 10 MLU value with respect to the traffic load for Nobel (a), Geant (b), and Germany50 (c) topologies considering HDF method

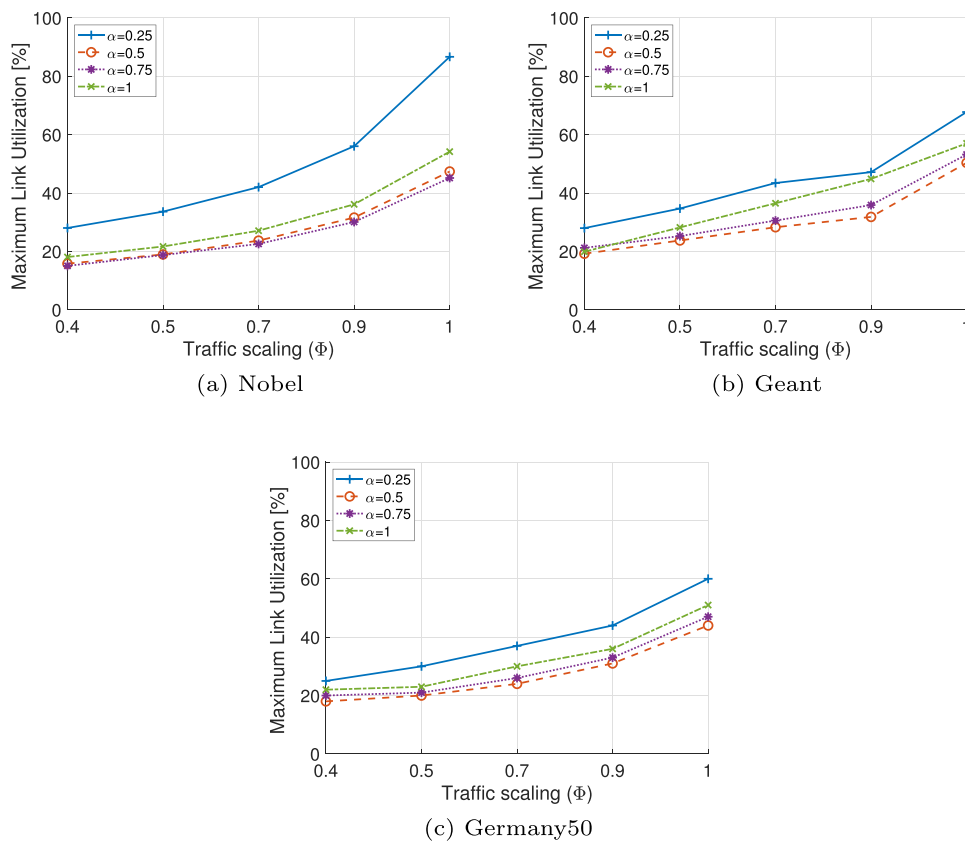


Fig. 11 Percentage of power savings obtained by L3A with respect to the traffic load for Nobel (a), Geant (b), and Germany50 (c) topologies considering HDF method

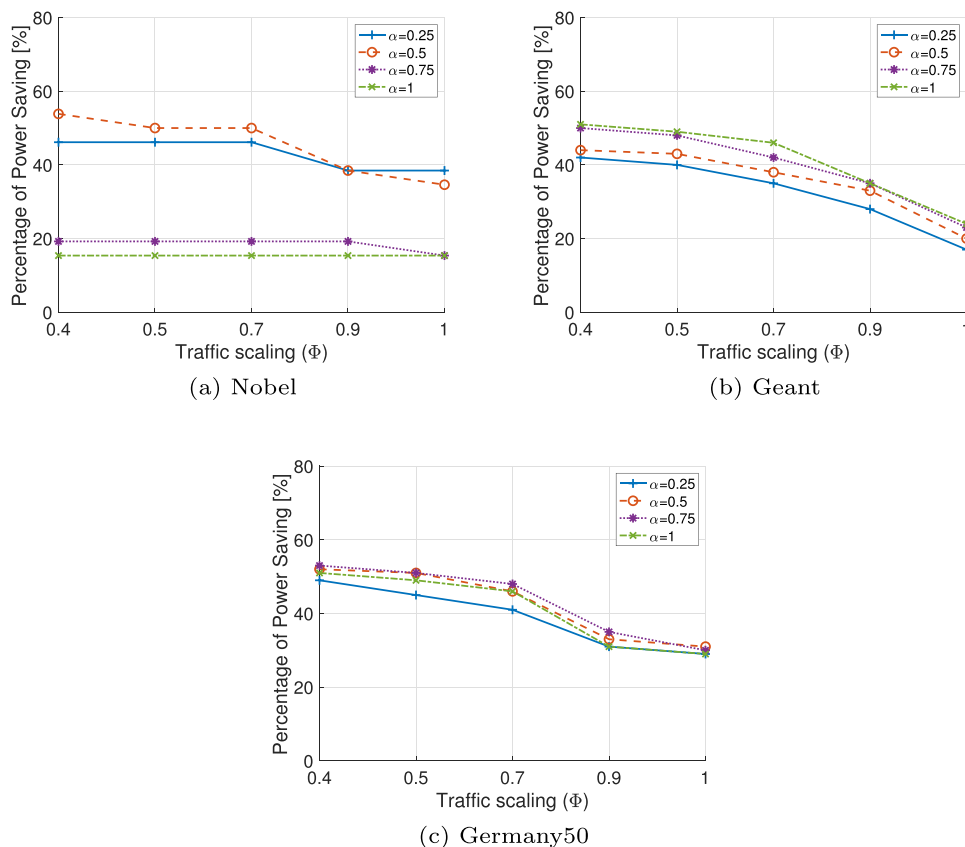
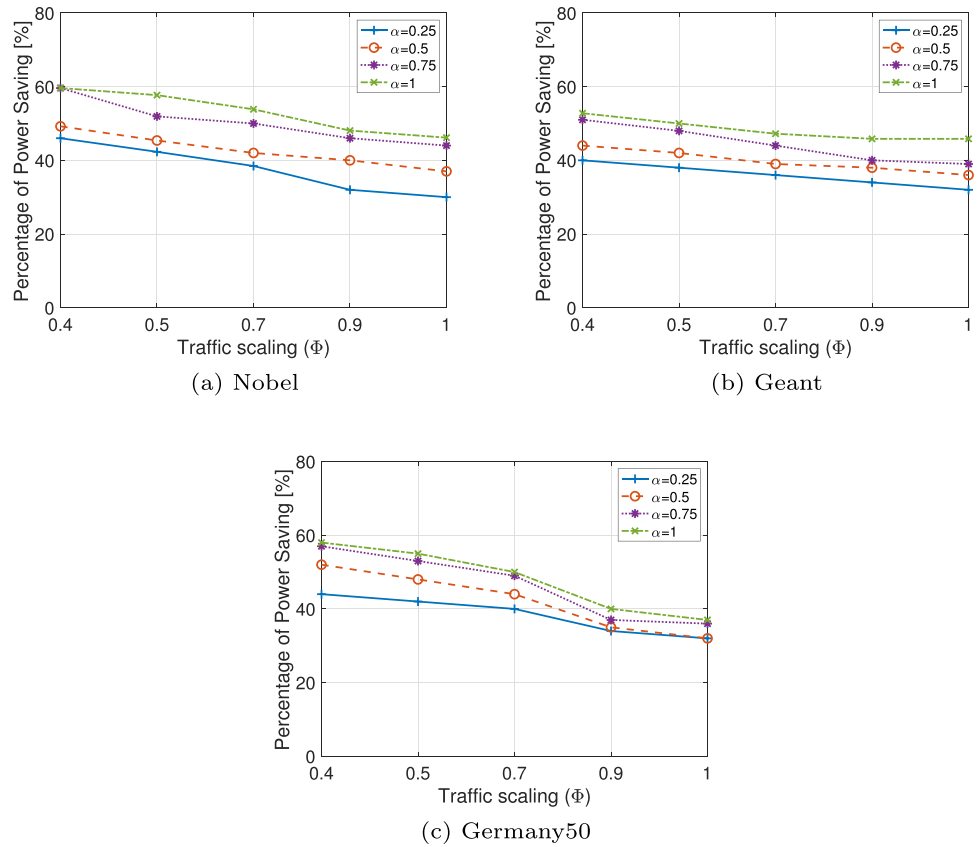


Fig. 12 Percentage of power savings obtained by GA-LSO with respect to the traffic load for Nobel (a), Geant (b), and Germany50 (c) topologies



case of the Germany network, GA-LSO slightly struggles with regard to L3A for a small percentage of SDN nodes ($\alpha = 0.25$) and a low traffic load. This may be caused by the large size of the chromosomes in large networks (each link is represented by a gene) and the flexibility for the routing given by the low traffic flowing throughout the network.

4.6 Comparison with other solutions

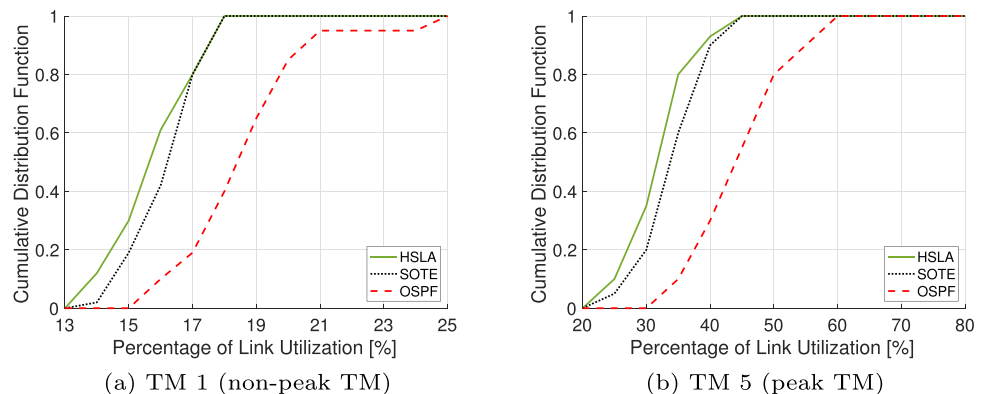
In this subsection, we first introduce two benchmark solutions and then we compare their results with those obtained by the proposed heuristic. Finally, the results of the proposed

solution are shown by means of a conducted performance analysis.

The first algorithm we focus on to highlight the benefit allowed by using HLSA is SOTE [27]. SOTE tries to minimize the MLU of the hybrid IP/SDN network by jointly optimizing OSPF’s weight settings for the whole network, thus balancing outgoing flows for IP equipment, as well as optimizing the traffic splitting ratio of flows aggregated at SDN nodes. However, SOTE does not consider energy efficiency optimization.

The second algorithm that is considered is HEATE [19]. HEATE’s focus is the minimization of the power

Fig. 13 Cumulative distribution function curves of link utilization for HSLA, SOTE and OSPF over Nobel topology considering TM 1 (a) and TM 5 (b)



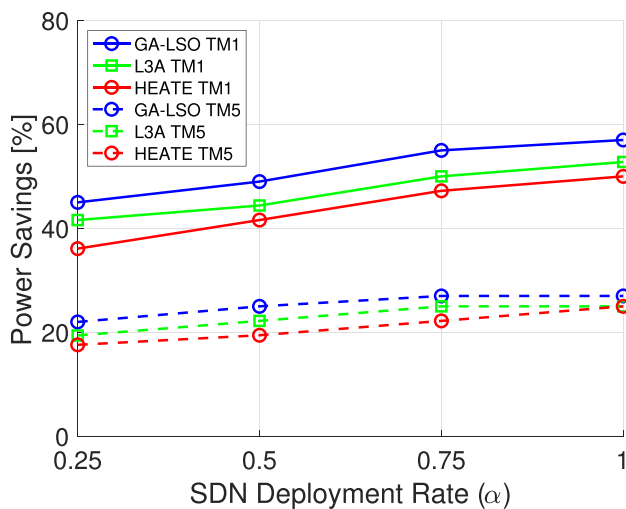


Fig. 14 Percentage of power savings achieved by GA-LSO, L3A, and HEATE over Geant topology considering TM 1 and TM 5

consumption of the hybrid IP/SDN network. Such a task is performed, similar to SOTE, by optimizing OSPF's link weight settings and SDN nodes' splitting ratio. However, HEATE does not consider the effects on the MLU caused by the minimization of energy consumption.

These two algorithms have been selected for comparison with the HSLA since they are both representative when facing the load balancing problem [27] and the energy consumption problem [19] in hybrid networks. Although works, such as [2] and [32], consider the joint problem of load balancing and energy efficiency in these scenarios, both of them are survey papers that review a set of works coping each type of problem ([2] considering also machine learning aspects), not the combined one. However, no proposals to compare with are provided.

Figure 13 reports the results for the first analysis, where the cumulative distribution function (CDF) curves of MLU are drawn for HSLA, SOTE, and OSPF. The deployment ratio is set to $\alpha = 50\%$, i.e., a half hybrid IP/SDN network, and two scenarios with the nonpeak TM (TM 1, Fig. 13a) and the peak TM (TM 5, Fig. 13b) are considered. As shown in both figures, the HSLA achieves the lowest MLU compared with the rest of the solutions. SOTE, which is specifically designed to minimize the MLU, presents slightly worse results for the case of low traffic (Fig. 13a), with an average of a 1% difference in the MLU with regard to HSLA in approximately 80% of the links. In the case of a scenario with high traffic (see Fig. 13b), the outcomes obtained by HSLA are better than those reported by SOTE, with an average MLU reduction of approximately 5% (note the difference in the x-axis between Fig. 13a and b). In general, the difference is that the HSLA allows for better load balancing in the less congested links with respect to SOTE for low traffic loads, while the gap is maintained

throughout all the links in cases of high traffic loads. Furthermore, the OSPF protocol presents worse results than the HSLA and SOTE protocols because its objective is not to minimize the MLU, but to find the shortest path among each source-destination pair.

To compare the power saving gains among GA-LSO, L3A, and HEATE, Fig. 14 shows the obtained results as a function of the SDN deployment rate for Geant topology. Two lines are reported for each solution for the peak and nonpeak TMs (TM 5 and TM 1). The figure shows that GA-LSO outperforms L3A and HEATE in all cases, with an average power saving gain of 7.7% with respect to HEATE and 4.2% when compared to L3A. In summary, the GA-based proposed solution presents better results than the ones obtained by ad hoc solutions that specifically cope a single objective.

5 Conclusion

The problems of load balancing and energy efficiency in hybrid IP/SDN networks are regarded by the state of the art as very relevant. However, the related literature treats them as completely separate problems, without considering the effects of powering off links on the MLU and vice versa. In this paper, we analyze the tradeoff between both problems. Starting from the opposite nature of the two optimization problems, a heuristic is proposed to jointly minimize the MLU and the network power consumption during the transition from IP to SDN networks. Through simulations on realistic network topologies, HSLA is compared with other state-of-the-art solutions, such as SOTE, which is intended only to minimize the MLU, while L3A and GA-LSO are compared to HEATE, which is aimed at solely minimizing the power consumption. Our proposed solution is shown to save more power than HEATE and outperforms SOTE in terms of link utilization. Moreover, it can provide up to 50% of MLU reduction and up to 60% of power savings. In the future, we plan to install the HSLA in a real SDN controller, such as Ryu or Faucet, to evaluate its impact on the network performance in a working test-bed.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work has been partially funded by the Diputación de Cáceres (AV-7), the project RTI2018-094591-B-I00 (MCI/AEI/FEDER,UE), the 4IE+ Project (0499-4IE-PLUS-4-E) funded by the Interreg V-A España-Portugal (POCTEP) 2014-2020 program, by the Department of Economy and Infrastructure of the Government of Extremadura (GR21133, IB18030), and by the European Regional Development Fund.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Cisco (2020) Cisco Annual Internet Report (2018–2023). <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf>. Accessed 2022-08-01
- Etengu R, Tan SC, Kwang LC, Abbou FM, Chuah TC (2020) AI-assisted framework for green-routing and load balancing in hybrid software-defined networking: proposal, challenges and future perspective. *IEEE Access* 8:166384–166441. <https://doi.org/10.1109/ACCESS.2020.3022291>
- Herrera JL, Polverini M, Galán-Jiménez J (2020) A machine learning-based framework to estimate the lifetime of network line cards. In: NOMS 2020 - 2020 IEEE/IFIP network operations and management symposium, pp 1–5. <https://doi.org/10.1109/NOMS47738.2020.9110455>
- Heegaard PE, Trivedi KS (2009) Network survivability modeling. *Comput Netw* 53(8):1215–1234. <https://doi.org/10.1016/j.comnet.2009.02.014>
- Bolla R, Bruschi R, Davoli F, Cucchietti F (2011) Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys Tutorials* 13(2):223–244. <https://doi.org/10.1109/SURV.2011.071410.00073>
- Chiaraviglio L, Mellia M, Neri F (2009) Reducing power consumption in backbone networks. In: 2009 IEEE international conference on communications, pp 1–6. <https://doi.org/10.1109/ICC.2009.5199404>
- Phillips C, Gazo-Cervero A, Galán-Jiménez J, Chen, X (2011) Pro-active energy management for wide area networks. *IET Conference Proceedings*, 317–3225
- Galán-Jiménez J, Polverini M, Cianfrani A (2018) Reducing the reconfiguration cost of flow tables in energy-efficient software-defined networks. *Comput Commun* 128:95–105. <https://doi.org/10.1016/j.comcom.2018.07.022>
- Galán-Jiménez J, Berrocal J, Herrera JL, Polverini M (2020) Multi-objective genetic algorithm for the joint optimization of energy efficiency and rule reduction in software-defined networks. In: 2020 11th international conference on network of the future (NoF), pp 33–37. <https://doi.org/10.1109/NoF50125.2020.9249089>
- Agarwal S, Kodialam M, Lakshman TV (2013) Traffic engineering in software defined networks. In: 2013 Proceedings IEEE INFOCOM, pp 2211–2219. <https://doi.org/10.1109/INFCOM.2013.6567024>
- Akyildiz IF, Lee A, Wang P, Luo M, Chou W (2014) A roadmap for traffic engineering in sdn-openflow networks. *Comput Netw* 71:1–30. <https://doi.org/10.1016/j.comnet.2014.06.002>
- Mendiola A, Astorga J, Jacob E, Higuero M (2017) A survey on the contributions of software-defined networking to traffic engineering. *IEEE Communications Surveys Tutorials* 19(2):918–953. <https://doi.org/10.1109/COMST.2016.2633579>
- Hong DK, Ma Y, Banerjee S, Mao ZM (2016) Incremental deployment of sdn in hybrid enterprise and isp networks. In: Proceedings of the symposium on SDN research. SOSR '16. association for computing machinery, New York, NY, USA. <https://doi.org/10.1145/2890955.2890959>
- Huang X, Cheng S, Cao K, Cong P, Wei T, Hu S (2019) A survey of deployment solutions and optimization strategies for hybrid sdn networks. *IEEE Commun Surv Tutor* 21(2):1483–1507. <https://doi.org/10.1109/COMST.2018.2871061>
- Amin R, Reisslein M, Shah N (2018) Hybrid SDN networks: a survey of existing approaches. *IEEE Commun Surv Tutor* 20(4):3259–3306. <https://doi.org/10.1109/COMST.2018.2837161>
- Giroire F, Moulhierac J, Phan TK (2014) Optimizing rule placement in software-defined networks for energy-aware routing. In: 2014 IEEE global communications conference, IEEE, pp 2523–2529
- Wang H, Li Y, Jin D, Hui P, Wu J (2015) Saving energy in partially deployed software defined networks. *IEEE Trans Comput* 65(5):1578–1592
- Huin N, Rifai M, Giroire F, Pacheco DL, Urvoy-Keller G, Moulhierac J (2018) Bringing energy aware routing closer to reality with sdn hybrid networks. *IEEE Trans Green Commun Netw* 2(4):1128–1139
- Wei Y, Zhang X, Xie L, Leng S (2016) Energy-aware traffic engineering in hybrid SDN/IP backbone networks. *J Comm Net* 18(4):559–566. <https://doi.org/10.1109/JCN.2016.000079>
- Saha N, Bera S, Misra S (2021) Sway: traffic-aware QoS routing in software-defined IoT. *IEEE Trans Emerg Top Comput* 9(1):390–401. <https://doi.org/10.1109/TETC.2018.2847296>
- Hopps C (2000) Analysis of an equal-cost multi-path algorithm. RFC 2992:1–8
- Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang Y, Kim DI (2019) Applications of deep reinforcement learning in communications and networking: a survey. *IEEE Commun Surv Tutor* 21(4):3133–3174. <https://doi.org/10.1109/COMST.2019.2916583>
- Lin T, Kuo C, Chang H, Chang W, Lin Y (2016) A parameterized wildcard method based on sdn for server load balancing. In: 2016 international conference on networking and network applications (NaNA), pp 383–386. <https://doi.org/10.1109/NaNA.2016.74>
- Agarwal S, Kodialam M, Lakshman T (2013) Traffic engineering in software defined networks. In: 2013 proceedings IEEE INFOCOM, IEEE, pp 2211–2219
- Ren C, Bai S, Wang Y, Li Y (2020) Achieving near-optimal traffic engineering using a distributed algorithm in hybrid SDN. *IEEE Access* 8:29111–29124
- Guo Z, Dou S, Wang Y, Liu S, Feng W, Xu Y (2021) Hybridflow: achieving load balancing in software-defined wans with scalable routing. *IEEE Trans Commun* 69(8):5255–5268
- Guo Y, Wang Z, Liu Z, Yin X, Shi X, Wu J, Xu Y, Chao HJ (2019) SOTE: traffic engineering in hybrid software defined networks. *Comput Netw* 154:60–72. <https://doi.org/10.1016/j.comnet.2019.03.008>
- Galán-Jiménez J, Polverini M, Lavacca FG, Herrera JL, Berrocal J (2021) On the tradeoff between load balancing and energy-efficiency in hybrid IP/SDN networks. In: 2021 12th international conference on network of the future (NoF), pp 1–9. <https://doi.org/10.1109/NoF52522.2021.9609876>
- Cianfrani A, Eramo V, Listanti M, Polverini M, Vasilakos AV (2012) An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks. *IEEE Trans Netw Serv Manag* 9(3):254–267. <https://doi.org/10.1109/TNSM.2012.031512.110165>

30. Ahn CW, Ramakrishna RS (2002) A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans Evol Comput* 6(6):566–579. <https://doi.org/10.1109/TEVC.2002.804323>
31. Orłowski S, Pióro M, Tomaszewski A, Wessäly R (2007) SNDlib 1.0—survivable network design library. In: Proceedings of the 3rd international network optimization conference (INOC 2007), Spa, Belgium. <http://sndlib.zib.de>, extended version accepted in *Networks*, 2009. Accessed 2022-08-01
32. Etengu R, Tan S, Abbou F, Lee C, Yusoff Z, Shahbe M (2019) Hybrid software-defined networking traffic scheduling: energy-aware load balancing perspective. In: Conference of open innovations association, FRUCT, pp 452–457. FRUCT Oy