Research Article

On adaptive multi-objective optimization for greener wired networks

Hatem Yazbek[1] · Peixiang Liu[1]

Received: 3 May 2020 / Accepted: 25 November 2020 / Published online: 8 December 2020

Abstract
In this study, we develop a novel adaptive Multi-Objective Optimization (MOO) algorithm to jointly minimize the Power Consumption (PC) and Maximum Link Utilization (MLU) of wired computer networks. This novel algorithm, based on Non-dominated Sorting Genetic Algorithm II (NSGA-II), is able to discover the optimal link weights configuration for wired networks based on live network traffic data. The goal is to minimize both PC and MLU when this optimal link weights configuration is applied to the network. The impact of the Internet and wired networks' growing power consumption on costs and environment continues to be a major concern of network operators. Our adaptive MOO scheme can be applied to any wired network to reduce its power consumption with acceptable load balancing to help achieve greener energy-efficient networks. In order to validate our solution, we run experiments on two different network topologies with various network traffic data to compare the performance of three different methods of updating the link weights: random, delta-weight, and hybrid. The experiment results show that all three methods can find optimal solutions to achieve both reduced PC and MLU. Compared to the default link weights configuration, the optimal solution found by hybrid approach is the best since it can reduce the PC by 35.24%, while reducing MLU by 42.86% for specific traffic pattern on Abilene network topology.

Keywords Multi-objective optimization · Energy-aware · Traffic engineering · Genetic Algorithm · Link weights configuration

1 Introduction

In this research, we attempt to minimize the network power consumption and achieve balanced network load at the same time using a Multi-Objective Optimization (MOO) method through monitoring the prevailing traffic changes and adapting the optimization algorithm accordingly. Vallet et al. implemented an online approach to optimize the OSPF weights in order to cover dynamic changing network traffic but they did not consider energy aware concern as they focused on reducing routing congestion [1]. Research to date with respect to energy aware network optimization relied on measuring network traffic and matrices that are not dynamic nor changing as described in [2, 3]. Many researchers have been utilizing Evolutionary Algorithms (EA) such as MOO for solving engineering problems that have more than one optimization objective, where there is no single optimal solution but rather a set of optimal solutions called Pareto front [4, 5].

Other approaches in the energy-aware network field focus on minimizing only the network power consumption and are able to find energy-efficient paths. However, using only those energy-efficient paths might cause oscillation and an imbalanced network since those paths are used to carry network traffic and may not always be energy-efficient anymore due to the heavy utilization. Our genetic algorithm based MOO approach minimizes not only Power Consumption (PC) but also Maximum Link Utilization (MLU) which provides an energy-efficient solution to wired networks without sacrificing the network load balancing.

✉ Hatem Yazbek, hatemy@gmail.com | [1]Nova Southeastern University, Fort Lauderdale, FL 33314-7795, USA.

We validated our approach by conducting experiments on two different network topologies with various network traffic data. The experiment results show that all three methods can find better link weights configurations to achieve reduced power consumption and smaller MLU. Compared to the default link weights configuration, our hybrid approach can reduce the PC by 35.24%, and lower the MLU by 42.86% for specific traffic pattern on Abilene network topology.

Our approach can be applied to any wired networks to help achieve network energy efficiency especially in networks managed by Software Defined Networking (SDN). SDN control layer controls packet forwarding among a network's SDN-enabled devices as well as the configuration and management of these devices. Our GA based MOO module can be implemented as a network control application similar to other applications such as routing, access control and load balance in SDN. The SDN controller gathers information from the network devices using protocol such as OpenFlow and forwards it to the GA-based MOO application. Our algorithms are then executed to search for the best configuration solutions based on this real-time information which are then returned to the SDN controller. The SDN controller can easily modify the link weight values it maintains and trigger the routing network control application to recalculate the flow tables for all the routers in the network. The OpenFlow control messages are then sent to those SDN enabled routers to modify the flow table entries, or enable/disable those routers.

In this paper, we first introduce the background and prior work in Sect. 2, and then we present the problem formulation in Sec. 3. In Sect. 4, we describe the approaches that we developed. Section 5 presents the performance evaluation based on simulation results, and finally Sect. 6 concludes the paper.

2 Background and prior work

2.1 Energy aware networks and traffic matrices

By 2020, European Telecoms are expected to consume around 35.8TWh of total network power annually, where the contribution of backbone networks out of total network power consumption will increase from 10 to 40% by 2020 [6]. Andrae et al. [7] also expected an aggregate growth rate of 7% per year, and calculated that ICT operation will rise to 21% of global electricity consumption by 2030. Reducing power consumption of network infrastructure and elements in the Internet has been the subject of recent research. Those studies focus on saving energy based on discovering network links and routers that are less utilized or under light traffic loads, directing traffic away from those devices and then putting them to sleep mode. A fair number of recent research works as shown in Energy-Aware Routing (EAR) by Bianzino et al. [8] have been dedicated to energy-aware traffic engineering. The energy efficiency issue has also become a high priority objective for wired networks and service infrastructures [9]. Because of the potential economic benefits and its expected environmental impact, reducing power consumption is becoming a major concern in wired networks [10, 11]. Therefore, green networking has been drawing a lot of attention in the last years, emphasized in surveys of [9, 10, 12, 13].

The network configurations can be re-computed in order to save energy [14]. The prime network infrastructure, such as routers, switches, and other devices, still lacks effective energy management solutions [15]. Reducing power consumption has been an important part of networking research with less focus on wired networks [16]. For tackling the increase in network power consumption, various approaches for managing the network load balance have been proposed such as GreenTE traffic engineering mechanism, Interior Gateway Protocol Weight Optimization (IGP-WO), and Green Load-balancing Algorithm (GLA) by [6, 15]. Green networking strategies were described in detail in the work of [10]. Bianzino et al. identified four classes of power-aware solutions: resource consolidation, virtualization, selective connectedness, and proportional computing [10]. The authors differentiated online solutions that act on runtime from offline solutions that act before runtime. However, doing so required frequent adjustment of network routing traffic status and raised the question of how often adjusting of the traffic matrix is required. A concept for measuring traffic, called traffic matrices was introduced in [15], where Zhang et al. showed that a small number of those matrices were enough to perform network analysis. They proposed a complementary approach to utilize power management at the network level, by routing traffic through various paths in order to adjust the workload on individual links or routers. They based their power saving assumptions on the fact that high path redundancy and low link utilization exist in today's networks. Traffic matrices and network measurement methods are crucial for understanding network behaviour, and for proposing effective solutions for saving energy. Researchers must be aware of the response time for waking the network device up after putting it to sleep, and also must account for traffic changes. One of the difficulties coming up with a valid solution is that traffic patterns change very frequently [17]. Dabahi et al. [18] presented a survey on green routing protocols where they identified and classified the main numerical metrics for comparing and evaluating the efficiency of sleep-scheduling. Bouras et al. [19] proposed a bi-level optimization

model where the upper level represents the energy management function, and the lower level refers to deployed multi-path routing protocol.

2.2 Network energy aware approaches and methods

In a latest survey Idzikowski et al. classified various approaches covering energy efficiency in core networks with respect to optimal formulations and heuristic solutions [20]. Most existing approaches rely on heuristic methods in order to compute those energy-aware critical paths, or calculate those links that can be put to sleep. This is because finding the optimal solution is a NP-hard problem as shown by Zhang et al. [15] and Vasić et al. [14]. Consequently, they recommended using pre-established paths that are constructed off-line in order to avoid long path calculation time. Although a power down approach was used in Vasić et al. [14], turning routers off was not the main concern as they identified a few energy-critical paths off-line, configured them in network devices, and redirected the traffic such that large part of the network entered low power state. Ultimately, they developed a new energy saving scheme that is based on identifying and using energy-critical path.

Concentrating network traffic on a minimal subset of network resources was addressed in [3]. In this approach, several backbone nodes got aggregated into one node and after finding an optimal solution for the reduced version of the network they had a method to revert the original nodes back. The optimal solution utilizing a heuristic approach and an Integer Linear Programming (ILP) solver was much more viable since the number of nodes was reduced. For solving energy optimization problems, most approaches established ILP and then designed comparable energy routing algorithms. Majumder et al. [21] and Lei et al. [22] chose newly active paths with high utilization to save energy. A modified version of OSPF protocol that reduces the number of active links by utilizing shared shortest paths trees was shown in [23]. This mechanism allowed sharing a Shortest Path Tree (SPT) between adjacent routers, so the total number of active links can be reduced and in a distributed way. The use of Genetic Algorithms in solving network energy aware problem is limited in literature, but one such approach was introduced in [6]. Francois et al. used GA to find the link weights for the joint optimization of load balancing and energy efficiency, and modified the existing traffic engineering scheme based on link sleeping operations in order to improve end-to-end traffic delay performance. Their work was based on sampling a few different traffic matrices (offline analysis), and found that IP networks have regular traffic patterns. ElasticTree was introduced by Heller et al. [17], which optimized the power consumption of Data Center Networks by turning off unnecessary links and switches during off-peak hours. ElasticTree also modeled the problem based on the Multi-Commodity Flow (MCF) model, but was focused on Fat-Tree or similar tree-based topologies. ElasticTree considered link utilization and redundancy when calculating the minimum-power network subset, and was implemented using OpenFlow. This approach leveraged the tree-based nature of these networks (e.g., a Fat-Tree) and at runtime computed a minimal set of network elements to carry the traffic. This is similar to our approach where we recomputed energy critical paths in order to reduce power, which was calculated online or at runtime as traffic changes.

Multi-Objective Optimization (MOO) is a part of evolution based algorithms. Evolutionary Algorithms (EA) are stochastic search methods that mimic the survival of the fittest process of natural ecosystems, that were introduced to optimize MOO problems [5, 24, 25]. MOO is used in solving difficult problems in several engineering applications such as energy savings, robotics, mechanical engineering, solar energy, thermodynamics, and others [4, 5, 26]. In the study of [27], a GA based approach that randomly searches for feasible network architectures is proposed. A network with M nodes and N links is presented as a chromosome with $M + N$ genes. Each gene can have a value from set {0, 1, 2} to denote different states of a node or link: inactive, active, or permanently active. In their approach, a chromosome is a candidate network configuration to be applied to the network. In MOO, a set of substitute optimal solutions are considered rather than a single optimal solution. When considering all objectives, we find those superior optimal solutions in the search space that are called Pareto Optimal (PO) solutions [4, 5, 28]. The primary target of MOO algorithms is to search an accurate approximation of the exact true PO solutions [25]. Lately, more attention is paid to environmental protection and to energy efficiency and economic profits, that are modeled as a MOO in real world problems [5]. Adapting the Software Defined Networking (SDN) approach in order to handle energy-aware routing and resource management for large scale Multiple Protocol Label Switching (MPLS) networks is described in [29]. They developed a controller using Pre-Established Label Switching Paths (PLSPs), which performs load balancing to minimize congestion in paths, and introduced a path virtualization concept. In the survey of [30] Assefa et al. conferred a study on the energy efficiency in SDN, and they presented numerous Traffic Aware Energy efficiency approaches. Our approach belongs to the Traffic Aware Energy efficiency approaches presented in this paper. The first-fit heuristic approach was used in [31]. Nodes and links are sorted using three criteria: Most-Power (MP), Least-Flow (LF), and Random (R). The goal is

to turn off the node/link which consumes the most power or carries the least flow.

Our preliminary research has been reported in [32] which established the initial research of using MOO for minimizing both MLU and power consumption, while using live traffic data. adaptive Multi-Objective optimization is also the topic of a PhD dissertation at Nova Southeastern University done by one of the authors of this article [33].

3 Problem formulation and description

In this section, we repeat the problem formulation described in detail in [32] for the sake of completeness. In order to find optimal solutions to energy-aware routing, and to find candidate links that can be turned off or put to sleep, most recent approaches considered by researchers rely on heuristic methods as in [16, 17]. The majority of researchers formulate the energy-aware problem as Mixed Integer Linear Programming (MILP) models [15]. Computing optimal solutions to energy-aware routing problem is an NP-hard problem [15]. In this research we found good solutions to a multi-objective function that models both PC and MLU using evolutionary optimization techniques [34]. The first objective is to reduce power consumption of the network, while the second objective is to reduce the MLU as part of achieving network load-balancing. The objective of energy reduction is achieved by utilizing only a set of shortest paths to carry the network traffic and thereby reducing the number of active links. Since fewer links are used to carry the traffic, the network link utilization is increased. We developed an algorithm that uses MOO and reduces network power consumption while maintaining network MLU below certain allowed threshold. In MOO, the objective functions are conflicting, and there are a (possibly infinite) number of Pareto-optimal solutions. Each solution that is chosen out of a set of Pareto-optimal solutions presents a different trade-off between MLU and power reduction. We want the links to be well utilized, but not to exceed an upper limit for MLU. On the other hand, the MLU lower limit is applied in order to reduce the number of solutions being generated. No solutions with an MLU lower than MLU lower limit are considered. The upper limit and lower limit of MLU are selected based on the network traffic and topology, and are pre-determined by the network operator.

3.1 Problem formulation

All the symbols used for problem formation are defined in Table 1. We model the network as a directed graph $G = (N, L)$, where N is the set of nodes (routers) and L is the set of links. We use a directed graph since the amount of network traffic flow can vary based on its direction, meaning a flow from node i to node j can be different from the flow from node j to node i. A link $l \in L$ from node i to node j can be put to sleep if there is no traffic on the link, and a node $m \in N$ or a router can be put to sleep if all of its links are asleep [15]. Given network topology G and demand volumes D for all SD pairs, the model determines network topologies and link weights matrix comprised by active links. Let K denote the total set of SD pairs, which has $|N|^2$ number of pairs. For every SD pair$(s, d) \in K$, there exists a traffic demand D^{sd} ,and for every link l from node i to node j there is a traffic flow f_{ij}^{sd} which is used as one of the decision variables as shown below.

Table 1 Summary of Model Notations

Notation	Description
Objective Functions	
$MLU\ [\%]$	Maximum Link Utilization
$PC\ [W]$	Power Consumption
Model Parameters	
$G(N, L)$	Directed graph with set of nodes N and set of links L
$P_m\ [W]$	Node power consumption of router m
$P_{ij}\ [W]$	Power consumption of link l from node i to node j
$u_{ij}\ [\%]$	Utilization of link l from node i to node j
$c_{ij}\ [bps]$	Bandwidth capacity of link l from node i to node j
$D^{sd}\ [bps]$	Traffic demand from node s to node d
$\alpha\ [\%]$	Maximum allowable utilization ratio of link capacity
K	Set of all $\|N\|^2$ Source-Destination (SD) pairs
Decision Variables	
x_{ij}	1 if link l from node i to node j is active, 0 otherwise
y_m	1 if node m is active, 0 otherwise
$f_{ij}^{sd}\ [bps]$	Traffic flow from node s to node d that traverses link l from node i to node j

Decision variables

- $\boldsymbol{x_{ij}}$: Binary decision variable that represents the power status (on/off) of link l from node i to node j. A value of 1 indicates that link l is active and 0 otherwise.
- $\boldsymbol{y_m}$: Binary decision variable that represents the power status (on/off) of node m (router). A value of 1 indicates node m is active and 0 otherwise.
- $\boldsymbol{f_{ij}^{sd}}$: Traffic flow from node s to node d traversing link l from node i to node j.

Objective functions

Joint optimization of network power using shortest paths and shutting down as many links and nodes as possible, while minimizing link utilization in order to achieve load balancing, can be formulated as a MOO problem with the following two objectives:

$$minimize\_PC = \sum_{(i,j)\in L} P_{ij}x_{ij} + \sum_{m=1}^{|N|} P_m y_m \tag{1}$$

$$minimize\_MLU = \max\left(\frac{1}{c_{ij}} \sum_{(s,d)\in K} f_{ij}^{sd}\right) \forall (i,j)\in L \tag{2}$$

Subject to following constraints

$$\sum_{j=1}^{|N|} f_{ij}^{sd} - \sum_{j=1}^{|N|} f_{ji}^{sd} = \begin{cases} D^{sd} & \forall s,d,i=s \\ -D^{sd} & \forall s,d,i=d \\ 0 & \forall s,d,i\neq s,d \end{cases} \tag{3}$$

$$\sum_{(s,d)\in K} f_{ij}^{sd} \leq \alpha x_{ij} c_{ij} \quad \forall (i,j)\in L,\ \alpha \in [0,1] \tag{4}$$

$$\sum_{(s,d)\in K} f_{ij}^{sd} \geq 0 \quad \forall (i,j)\in L \tag{5a}$$

$$x_{ij} = 1 \quad iff \quad \sum_{(s,d)\in K} f_{ij}^{sd} > 0 \quad \forall (i,j)\in L \tag{5b}$$

$$y_m = 0 \quad iff \quad \left(\sum_{(s,d)\in K}\sum_{(i,m)\in L} f_{im}^{sd} + \sum_{(s,d)\in K}\sum_{(m,j)\in L} f_{mj}^{sd}\right) = 0 \tag{5c}$$

$\forall m \in [1,|N|]$

Equation 1 represents the first objective of MOO which minimizes the Power Consumption (PC) by moving traffic to shortest paths (fewer devices to turn on thus reducing power). We measure the power consumption of the network as the total power consumption of active links and nodes at a specific time snapshot, and periodically sample the network power state after we execute our MOO. Equation 2 represents the second objective of our MOO which minimizes the MLU of the network in order to achieve load-balancing. The utilization of each link is calculated as the sum of all network flow through link *l* flowing from node i to node *j*, divided by the link's specified capacity. Equation 3 represents the standard flow conservation constraint. Equation 4 determines that whenever a shortest path topology is used, all active links should have their utilization below a specific threshold α, and forcing the flow to 0 if the link *l* from node i to node j is powered off. So we must ensure that when a set of links is put to sleep mode, the MLU does not exceed the link capacity thresholdα; when link is active and it is 0 when the link is sleeping. The flow on link *l* from node *i* to node *j* is the sum of all flows for every SD pair that traverses link*l*. The flow f_{ij}^{sd} exists only if the decision variable x_{ij} is 1. The SD pair from node *s* to node *d* is a subset of all SD pairs$\in K$, which go through link*l* from node*i* to node*j*. Equation 5a states the traffic flow carries a value equal to or greater than zero, while Eq. 5b ensures that the link *l* from node *i* to node *j* is operated if and only if its flow does carry non-zero traffic, and thusx_{ij}=1. Equation 5c constrains node *m* to be inactive if and only if the sum of all traffic flowing through it is zero, and thus y_m= 0. Both x_{ij} and y_m are binary decision variables that can be set to 0 or 1 values.

3.2 Descriptive example of network energy and MLU MOO algorithm

Figure 1 depicts the basic concept of link weight optimization for both minimizing power consumption and

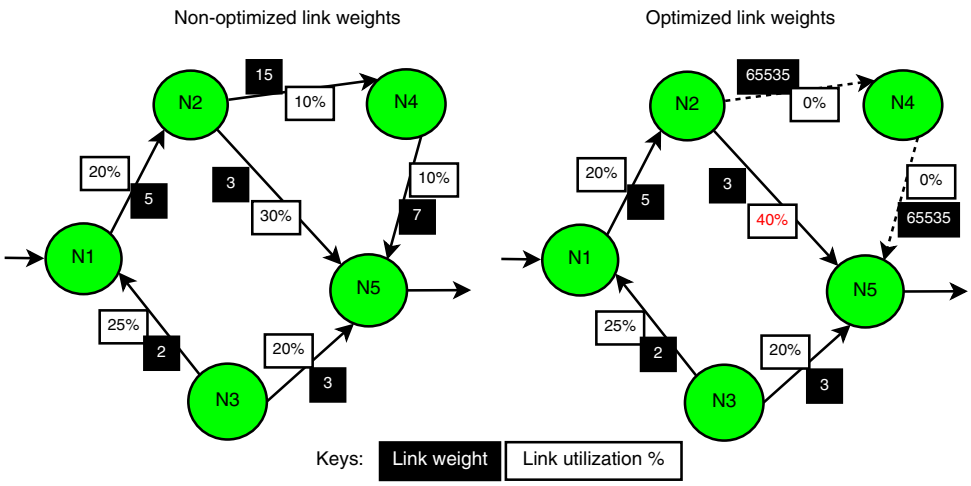


Fig. 1 Descriptive example for reducing network energy by topology link weights optimization

achieving load balance. We use a small example of network topology in Fig. 1 assuming link capacities of all links are at 100%, and that MLU must be below 50% (α) as an example. We show a weighted directed graph $G(N, L)$ where we have 5 nodes ($|N| = 5$), and 8 links ($|L| = 8$), with the indicated link utilization and link weight settings for each link. These link weights were selected for the sake of describing the optimization of link weights in order to save energy, while not exceeding the MLU limit.

The traffic from N1 to N5 can go on multiple paths, through N1-N2-N5, N1-N3-N5, or N1-N2-N4-N5. Clearly, we can shut down links N2-N4 and N4-N5 without causing the network topology to lose full connectivity. We first consider the case where a set of link weights are not optimized as in the left side of Fig. 1 where the maximum link utilization of 30% is currently found at link N2-N5. The case where the link weights are optimized is depicted on the right side of Fig. 1. Whenever we want to minimize the PC of this network topology, we select shortest paths and we increase their utilization assuming that we do not exceed its allowed MLU. Therefore, the algorithm does not select links N2-N4 or N4-N5, which are the least utilized links. Our algorithm uses a special link weight of 65, 535 to represent that the link is not utilized and it should be put to sleep. The MLU is increased, since N2-N4 and N4-N5 are put to sleep, and their traffic demands is routed along the other path of N2-N5. Evaluating the maximum link utilization function should yield 40%, which is found on link N2-N5. The PC is calculated based on Eq. 1, which is reduced since a router and two links are put to sleep.

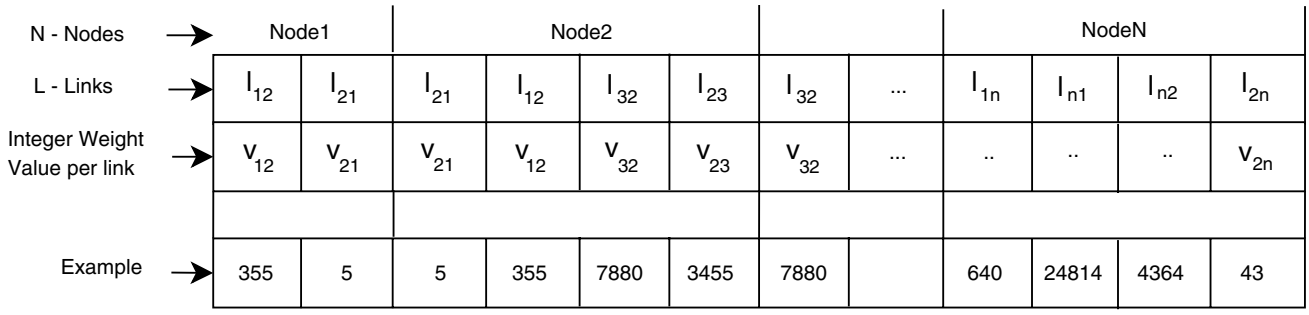
The example above shows that adjusting the link weights has allowed two links to go to sleep by directing traffic from under-utilized links to shortest paths, while not exceeding the allowable MLU of the network.

4 Methodology

In this section, we repeat the methodology described in detail in [32] for the sake of completeness. The study builds upon the work of [6] by using adaptive multi-objective optimization methods based on real network traffic. We developed methods to model the network inputs, and created Python code program that implements the objective functions to minimize the PC and MLU of the network. In our approach, Energy-aware traffic engineering is considered, along with the traffic topology and traffic flow. We proposed a GA-based algorithm called Multi-Objective Genetic Algorithm (MOGA) which uses MOO that reduces network power consumption while maintaining network MLU below a certain allowed threshold. The algorithm is based on ranking and selecting the population fronts by using two additional specialized multi-objective operators called non-dominance technique and a crowding distance [34]. Optimal solutions to a multi-objective problem are non-dominated solutions, known as Pareto-optimal solutions, where depending on the optimization case, one point is preferred over the other [35]. The MOGA uses a Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm with customized mutation, and crossover operators. NSGA-II works in a similar way as traditional genetic algorithms, but we chose it because it preserves elitism and diversity of the solutions space and has low computational complexity. NSGA-II is a multi-objective genetic algorithm that was first proposed by Deb. et al. [25]. NSGA-II finds a Pareto optimal front of solution space to minimize both PC and MLU of the network. In this Genetic Algorithm (GA), the solution is encoded through a chromosome, which is created from a number of genes equal to the number of network links. We introduced several more GA customized operators of crossover and mutation. These customized operators should further enhance NSGA-II basic operators and make the search in the Pareto-optimal solution space more efficient.

4.1 Genetic algorithm solution encoding, and scheme overview

Each chromosome has L genes where L is the number of links. Each gene represents a link weight and is represented as an integer value within a range of [1,65535], which is part of the link weights vector $w=(v_1, v_2,\ldots,v_L)$, where $v_i \in [1,65535]$ for each link $i=1, 2, \ldots, L$ [36, 37]. These link weights are used by routing algorithms to determine the paths being used to carry the network traffic. The network operator can tune the integer link weights, in order to compute shortest paths for carrying the network traffic [16]. Using Open Shortest Path First (OSPF), the network operator assigns a weight to each link, and shortest paths from each router (node) to each destination node are computed using the weighted cost function of the links [37]. The value of 65, 535 is the maximum value that OSPF allows [37], and in our approach, the maximum link weight value of 65, 535 is assigned to a link to indicate that it is supposed to be put into sleep mode and is not used to carry network traffic. However, the selection mechanism is up to the network operator, as another scheme or range of link weights can be used to enforce putting links into sleep mode. Each chromosome has two recognizable fitness functions in the NSGA-II algorithm that represent Eqs. 1 and 2, respectively. Figure 2 shows the encoding of a chromosome and how each gene represents an integer value that corresponds to a link weight. The higher the value of a link weight the lower the probability that the network operator will select this link for carrying traffic.



Note: Each gene can have a random integer weight value from [1, 65535].

Fig. 2 Network links mapping to chromosome encoding

Each node is connected to other adjacent nodes based on the given network configuration, and not necessarily to all other nodes. Figure 2 shows an arbitrary example of nodes connectivity to other links. For a given chromosome, both fitness functions are evaluated numerically to obtain a PC value and a numerical output of MLU. Our MOO module takes several inputs: a chromosome with link weights, network graph G, network demands D, and outputs two numerical values corresponding to two objectives: network PC and MLU. Calculating the objective functions is the most expensive process computationally since we have to repeat it for every GA generation. In each GA generation we evaluate the two objective fitness functions for every chromosome (solution) in the GA population. GA stops when the best solution has not improved in the last *gamax_num* number of generations. Our approaches rely on sampling traffic changes and matrices in a continuous fashion. The link weights in the best solution discovered by GA are used by OSPF to recalculate the paths being used to carry the current network traffic. The link utilization is then measured after we route the network traffic on top of the changed network with updated link weights. Thus the resulting chromosome with the optimal solution represented by different values of its genes is eventually mapped to a particular traffic routing solution. Our MOO uses adaptive strategies that optimize two objectives of minimizing PC and MLU.

4.2 Traffic engineering schemes and approaches

All of the developed approaches in this research are based on MOO. Network operators employ TE schemes in order to perform load-balancing. All approaches rely first on certain energy aware TE schemes that are based on sorting criteria or rules for the nodes and links [3, 38]. The sorting criteria for nodes that we consider for this work are shown in Table 2. They include Random (R) and Most-Power (MP) for nodes, Least-Flow (LF) for both nodes and links, and Least-Link (LL) for links. In the following sections, we describe three approaches that use the sorting criteria listed in Table 2.

Table 2 Criteria for link/node sleeping

Criteria	Applies to	Description
LF	Nodes & Links	Consider smallest amount of traffic
R	Nodes & Links	Random order
MP	Nodes	Consider highest power consumption
LL	Links	Sort link weight values monotonically

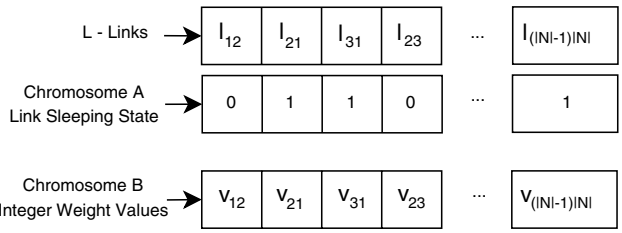


Fig. 3 Dual chromosome approach

4.2.1 Approach 1 (random MOGA-I)

In the first approach, the algorithm concurrently considers both the link sleeping scheme, and the link weights. It uses a Random (R) method to select the Link Sleeping State (LSS) value for each link, which indicates whether the link is active or asleep. In this approach the network measurement and initial or current link weights are not considered, and the MOGA-I algorithm finds optimal solutions to minimize both fitness functions defined in Eqs. 1 and 2. Two kinds of chromosomes as depicted in Fig. 3. are used in MOGA-I. One chromosome uses binary encoding (0,1) for each link *l* from node *i* to node j, so it includes *L* genes where every gene has a LSS value. The

second chromosome uses Integer Weights Values (IWV) per link as shown before. Each gene can have a random value $v_i \in [1, 65535]$, for i=1, ⋯ L.

The MOGA-I algorithm generates both chromosomes randomly, and it uses GA operators such as crossover and mutation to get optimal solutions. In each iteration, the values in those chromosomes are used to reconfigure the network. Based on the new network topology and updated link weights, our multi-objective optimization system uses OSPF routing and Equal Cost Multiple Path (ECMP) to calculate the shortest paths to carry the injected traffic flow. The PC and MLU are then calculated to evaluate the quality of the solutions represented by those chromosomes. Decision variables are obtained by MOGA chromosome A, while y_m is determined by x_{ij}: if links flowing into and out of a certain node m are all inactive or asleep, the node m is put into sleep. PC fitness function is easily calculated based on x_{ij} and y_m, and the power information of associated nodes and links. Consequently, new network flows f_{ij}^{sd} are obtained as the output of the network system along with the numerical value for another fitness function: MLU.

4.2.2 Delta weight approach 2 (delta MOGA-II)

In this approach, we use the values of the link weights as configured in the current network configuration as our initial link weights. In order to minimize oscillations in the network when applying new link weights and configuring new traffic flow matrix, we rely on changing link weights gradually based on the relatively small difference in each link weight value denoted as Δw(delta weight). We must make sure that the absolute value of $\Delta w$is carefully selected as it has to be big enough for the network operator to decide to change the traffic flow through that link. On the other hand, Δwshould not be too big in order to prevent oscillations in the network.

The MOGA-II algorithm described in Fig. 4 is a two-step solution. In the first step, the LSS values of the chromosome are pre-calculated based on the current network configuration. Pre-calculation of LSS values uses least-flow (LF) criteria for determining the value for each link. In the second step, the MOGA-II algorithm finds optimal solutions for minimizing objectives PC and MLU described in Eqs. 1 and 2, respectively. This MOGA-II algorithm also uses chromosome B in Fig. 3, but its link weights are generated in a different way. The IWV values of a chromosome are generated based on the IWV values of another existing chromosome by adding an incremental value, Δw. This incremental value is chosen from a limited range of values. Assume w_{ij}^{l} is the initial weight value for the link l from node i to node j, the differential weight value Δw for this

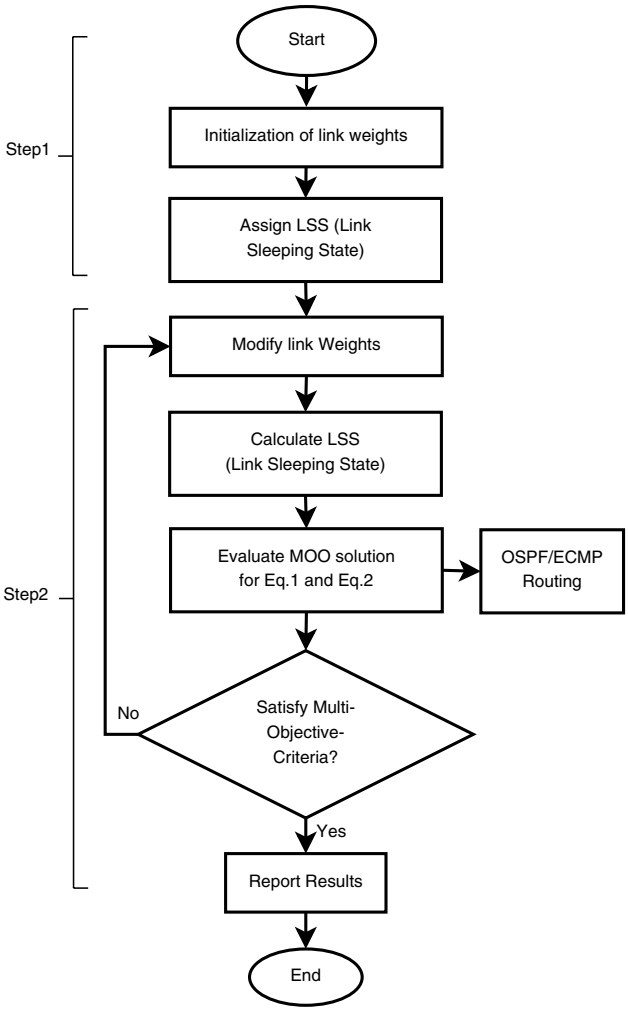


Fig. 4 MOGA-II two step algorithm

link is what we need in order to make the network operator change traffic through this link, either by reducing or by increasing its current link weight. We use Δw_{ij} to denote $\Delta w$for the link l from node i to node j at time T. Eventually, MOGA-II should select weight value $w_{ij}$for each gene from a value range described in Eq. 6 below:

$$w_{ij} \in [(w_{ij}^{l} - \Delta w_{ij}), (w_{ij}^{l} + \Delta w_{ij})] \quad \forall (i,j) \in L \tag{6}$$

The value of Δw_{ij} is carefully selected based on inputs from the network operator and based on the OSPF algorithm. Those inputs specify the value needed for deviating traffic from a certain link. Furthermore, modification of the link weights after we evaluate MOO (picking the "No" branch of "Satisfy Multi-Objective Criteria" in Fig. 4) is done by increasing the weight of least loaded links (LF) by Δw_{ij} (make them sleep), or by increasing the weight of other links by Δw_{ij} (deviate traffic flow). The parameter of

Δw_{ij} is re-computed and the updated new link weight is represented as follows:

$$w_{ij} = w_{ij}^{C} + \Delta w_{ij} \quad \forall (i,j) \in L \tag{7}$$

$$\Delta w_{ij} = (w_{max} - w_{ij}^{C}) \quad \forall (i,j) \in L \ \ if \ \ x_{ij} = 0 \tag{8}$$

Note that w_{ij}^{C} is the current link weight value for the link *l* from node *i* to node *j,* and w_{max} denotes the maximum link weight value in the current network configuration as evaluated by MOO. In the problem formulation it was indicated that if x_{ij} is 1 then the link is active, otherwise if x_{ij} is 0 then it is inactive or asleep. If the new link is still active, then based on Eq. 7 its weight w_{ij} is updated by adding Δw_{ij} to the current link weightw_{ij}^{C}. Otherwise if the new link is asleep then based on Eq. 8 its weight of w_{ij} is set to w_{max} and the link *l* from node *i* to node *j* remains at sleep state. As noted, our main goal is to switch off the maximum number of links for energy saving and then spread the traffic as evenly as possible on all remaining active links.

This approach of using Δwalong with the MOGA algorithm can reduce the computation time of the algorithm since it uses a smaller search space. The only drawback of this Δw approach is that the probability of being stuck in a locally optimal region is higher with this method compared to the approach 1 (Random MOGA-I) above.

4.2.3 Hybrid approach 3 (hybrid MOGA-III)

This approach combines Random MOGA-I and Delta MOGA-II described above, in which the Random MOGA-I is executed first, for a certain number of generations. The solution found by Random MOGA-I in the limited number of generations is then chosen to be used by Delta MOGA-II as the start solution to continue searching for better solutions. The main advantage of this hybrid or mixed approach is that it has the good features of both Random MOGA-I and Delta MOGA-II. Random MOGA-I yields better solutions but lacks stability and has more oscillations due to the random approach it takes to search for the best solutions. On the other hand, Delta MOGA-II has much less oscillations on the network traffic since it modifies the link weights by an incremental value from a limited range. Due to the smaller searching space, Delta MOGA-II is able to converge in a shorter time although the solution it finds might be a local optimal solution. The idea of Hybrid MOGA-III is to first run Random MOGA-I for $N1$generations to get a suboptimal solution, then switch to Delta MOGA-II using the suboptimal solution as the start point to search for the optimal solution. Doing so, Hybrid MOGA-III is able to find the optimal solution as good as that discovered by Random MOGA-I in a much shorter time due to the smaller searching space and a much better starting point used being by Delta MOGA-II in the second stage.

4.3 Adaptive GA genetic operators

All three approaches described above use GA or MOGA to search for good solutions using mutation and crossover operators. In this work, the GA operators are adaptive, as they depend on the network configuration inputs of topology $G = (N, L)$, traffic matrix *D*and the initial set of current link weights. The fitness functions depend on those inputs and can have a different output every time those inputs change. Mutation and crossover operators take one or two solution candidates (chromosomes) and alter the values (IWV) of one or more genes in the chromosomes [6]. Both operators are customized to fit the optimization problem that we have, by considering both LSS and IWV values for every network configuration represented by *G*,*D*, and initial network link weights.

4.3.1 GA crossover operator

In GA, a crossover operator takes two chromosomes (i.e. two solutions) in the current population and swaps their genes (i.e. link weights) with each other in order to generate two new offspring chromosomes. We use the most common crossover operator: the two-segment crossover. In addition, a new LSS aware crossover operator is defined which performs a multiplication function (AND) on the LSS values of the links that are associated with the two parent chromosomes. This crossover operator may urge link sleeping to multiply through the population, since one parent chromosome can alter the corresponding gene in the other parent chromosome to its own value if it represents a sleeping link. Figure 5 shows an example of parent chromosomes where each parent is composed of genes which represent the sleeping status of the network links. The LSS values for the two parents are shown, and if the link is sleeping then the value of LSS is "0", otherwise the link is active and the value of LSS is "1". This customized crossover operator is designed so that one parent chromosome can replace the corresponding gene of the other parent chromosome if its LSS value is "0", which represents that the link is sleeping. The resulting child as shown in Fig. 5 has the AND of the LSS values from the two parent chromosomes as the LSS value of its own gene. The arrows between the two parent chromosomes show the direction of the change when the LSS change occurs.

4.3.2 GA mutation operator

Another GA operator that we have customized is the mutation operator, which modifies one or more genes (i.e. link

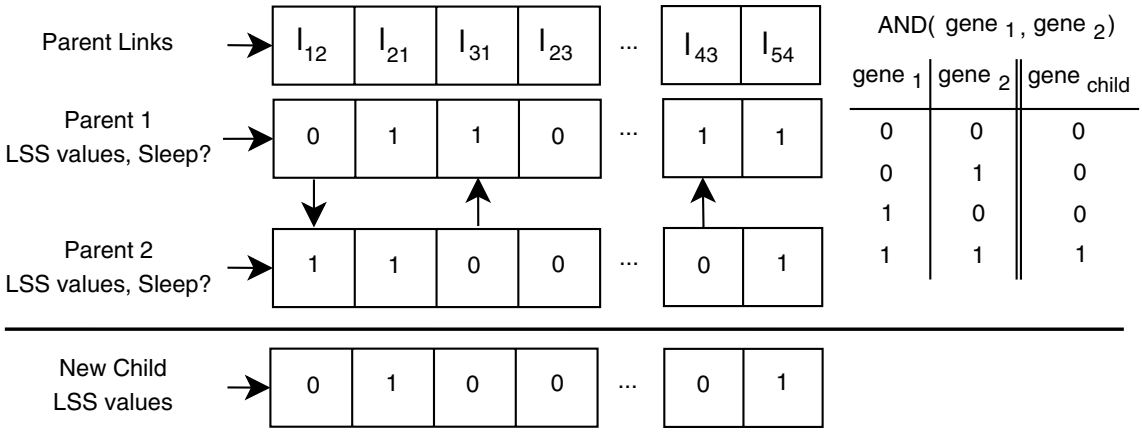


Fig. 5 LSS aware crossover operator

weights) in the chromosome. We associate the mutation probability of each gene directly with the link utilization and MLU in the network, such as $P_{ij}^{mutation} = \frac{u_{ij}}{MLU}$ as described in the context of the work in [6]. Whenever the link is more utilized, its mutation probability becomes higher. This way a link that has high utilization will have its link weights increased, and thus less likely to be selected for the network traffic routing. Anytime we generate a chromosome via mutation, the mutation can be done either by the new customized mutation method or the random one, but not by a mixture of both for the same chromosome. If the new customized mutation method is used, each gene is mutated with the probability linked to the utilization of the link that the gene represents, and the link weight changes according to the function in Eq. 9 below:

$$w_{ij} = \begin{cases} w_{ij}^{C} + \Delta w_{ij} & \forall (i,j) \in L,\ if\ u_{ij} < u_{lo}\ or\ u_{ij} > u_{hi} \\ w_{ij}^{C} - \Delta w_{ij} & \forall (i,j) \in L,\ if\ u_{lo} < u_{ij} < u_{hi} \end{cases} \tag{9}$$

In Eq. 9, we compare the link utilization value to pre-established limits for maximum link utilization u_{hi} and minimum link utilization u_{lo}. Highly utilized links should have their link weights increased in order to reduce MLU, so that there is more chance that traffic goes away from them. Low utilized links should have their link weights increased as well so they go to sleep. Medium utilized links should have their link weights decreased in order to take more traffic. Thus, this method adaptively tunes mutation rate according to the specific inputs from the network.

5 Performance evaluation and results

The goal of this research was to develop adaptive strategies for a multi-objective genetic algorithm such that both objectives of reduced network power consumption and load balance are achieved. In order to achieve this goal a new approach was introduced and implemented using network real time traffic that was generated by Multi Generator (MGEN). This research investigated the use of MOO in order to find good solutions for reducing PC while maintaining network MLU below certain allowed threshold.

The network system and methodology is shown in Fig. 6, where network topology and traffic flow generated by MGEN is fed into the network emulator of Common Open Research Emulator (CORE). CORE is a computer network emulation tool from Naval Research Laboratory (NRL), and it is used to emulate the network and capture the network topology [39]. We measure the MLU and PC in each iteration and feed them back to the MOO algorithm, which is implemented using our Python program

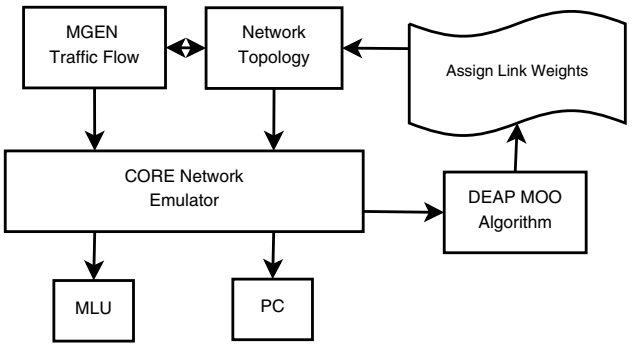


Fig. 6 Diagram of CORE and algorithm for setting link weights

that uses Distributed Evolutionary Algorithms in Python (DEAP) for the multi-objective optimization [40]. DEAP is a novel evolutionary computation framework that can be used for Genetic Algorithms (GA), and Multi-Objective Optimization such as NSGA-II [34].

In order to test our methodology in real time and using a flow based traffic we used a network emulator that can resemble a real computer network running on actual routers and links. We used MGEN, which is both a command line and GUI traffic generator that was developed by the US Naval Research Laboratory, to generate real network traffic flows. For calculating the MLU fitness function, we use a Python script, which considers the resulting CORE simulated traffic generated by MGEN. For operational network topologies, we use the widely used SNDLib to provide the network topologies of backbone networks such as Abilene and Polska [41]. The second fitness function of the PC is calculated in parallel by a Python utility which extracts and reflects the values of the network flow monitor. Both fitness functions are fed back to a Python program which implements the proposed MOGA algorithms and figures out the best solution. CORE uses Quagga/Zebra software for managing the OSPF and routing parameters. Furthermore, using Quagga shell "vtysh" we can configure the OSPF link weights on the fly while CORE network emulation process is running. Quagga routing software enables OSPF-V2 protocol with Equal Cost Multiple Path (ECMP) multicast methods.

We start the CORE simulation by using default link weights, and once the traffic is established we start running the MOO algorithm that yields new link weights for all links in network topology. In every iteration, we collect the resulting MLU and PC and after several iterations, DEAP keeps track of HOF (Hall of Fame) best Pareto optimal front solution so far in the algorithm. Those resulting MLU and PC are also used as an input for MOGA mutation and crossover functions as described earlier in our methodology. At the end of the simulation, the Python program using DEAP gives the multi-objective optimal solution for both MLU and PC from the Pareto optimal front.

5.1 Network and traffic description

Performance of the three various approaches discussed in previous methodology Sect. 4 was first evaluated on network topology of Abilene Orlowski et al. [41] which has been used for academic research since 2004. This topology is shown in Fig. 7a and it includes 11 nodes and 14 links. The power metrics used in this study are based on the same information used for GreenTE algorithm in the work of Zhang et al. [15], and based on Cisco line cards information. We used 0.6 kW for each link and 10 kW for each node. For the sake of simplicity, we assumed that all network links have the same traffic capacity. Both links and nodes were considered when calculating the total PC of the network consisting of links and nodes. We picked three network traffic data sets from Abilene recorded data, which correspond to 5th of June, 5th of August, and 5th of September. The data sets depict the total traffic generated in each of those days per minute of time. We then translated the data sets into MGEN packet data format using Poisson distribution. The data sets were obtained from SDNlib and also verified to be the same from the work of [42]. It shows how the arrangement of data for each node versus the rest of the nodes as a topology matrix. We translated this format of flow description to MGEN tool format, by specifying the flow "from node *i* to node *j*", until all aggregate flows were combined and fed to CORE emulator. CORE emulator takes as input the converted MGEN traffic below, the network topology and configuration of OSPF, and Quagga routing flow. The second network topology of Polska contains 12 nodes and 18 links as shown in Fig. 7b earlier. We could not find appropriate released data sets for this network topology, and as indicated in the research work of [43], we created similar traffic flow, and by using MGEN tool we developed a method to aggregate the traffic flow in order to create similar load balance and PC used in Abilene network topology. For power metrics, we used similar PC per link and node as discussed in Abilene topology.

5.2 Simulation results

Extensive simulations were carried out in order to compare the performance of the various approaches and to verify the potential for saving power while keeping MLU at acceptable level. We validated all approaches including a comparison of MLU and PC results which showed the best approach results. All approaches yield new link weight values, which are flooded in the network using control messages. The routers then re-compute the shortest paths (OSPF) and update their forwarding routing tables; this may take a few seconds until all routers stabilize their shortest paths [44].

5.2.1 Δw simulations for the Δw approach

For the delta weight approach, we used Δw in order to determine how much each link weight could be varied and obtained better solutions beyond the initial link weights that we started with. We varied the Δw in order to find the best value that yielded good solutions for our multi-objective optimization algorithm. Figure 8 describes how MLU and PC vary as we change the Δw value from 1000 until 12500 on Abilene topology. We can see that the best MLU and PC multi-objective solutions are obtained

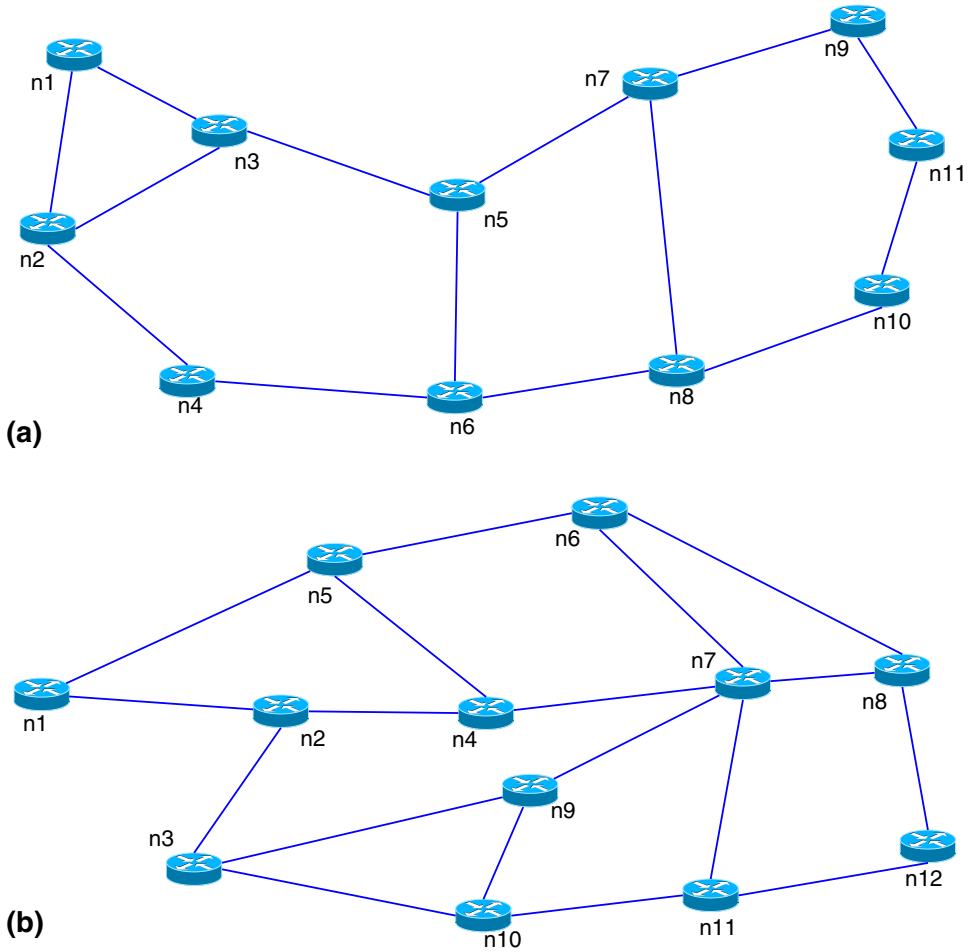


Fig. 7 **a** Abilene topology, **b** Polska topology

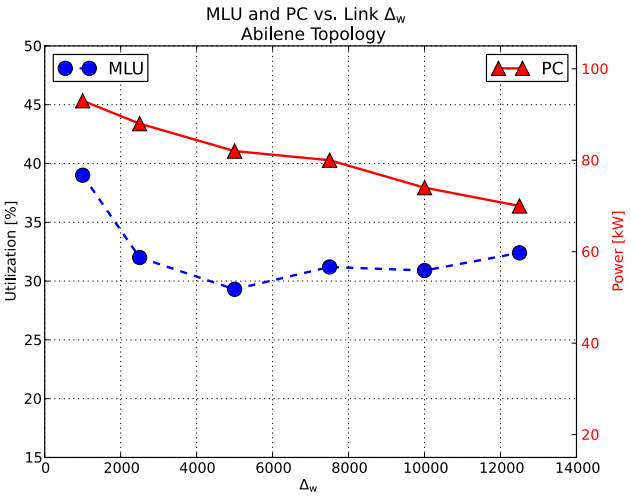


Fig. 8 MLU and PC versus link Δw Abilene topology

when we increase Δw higher until we reach a value of 5000 where MLU is lowest. For PC, it keeps improving slightly as Δw increases beyond 5000 until 12500. Considering both objectives, we get that Δw of 5000 produces the best results.

For Polska topology, we get similar results when we vary the Δw value, with different levels of PC and MLU numbers. Figure 9 shows the results of best MLU and PC values that we obtained when we varied the Δw parameter with medium traffic on Polska topology, and as we can see from the graphs, we do get the best solutions around Δw of 5000, considering both objective functions.

5.2.2 Abilene network topology results

Figures 10 and 11 show the results of PC and MLU, respectively. We present 4 types of data results, where the first type is the default, and uses the default link weights that match the network topology. The other three types match the approaches of random, delta, and hybrid as discussed in the previous section.

Power consumption results from those three approaches is mainly affected by the assumption that a node can be put to sleep entirely if all of the incoming and outgoing flows through the node are negligible. When a

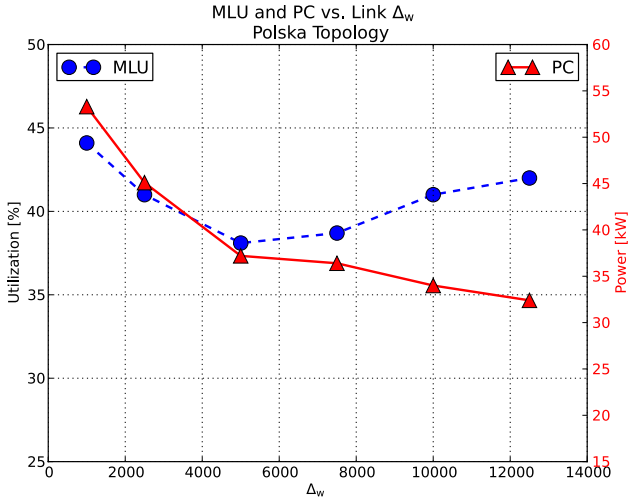


Fig. 9 MLU and PC versus link Δw Polska topology

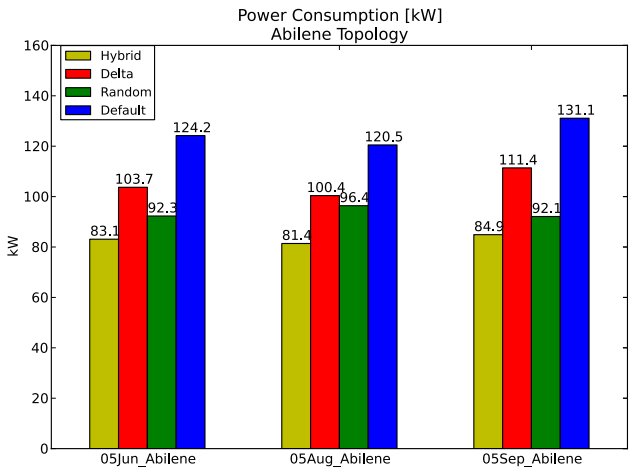


Fig. 10 PC versus approach, abilene topology

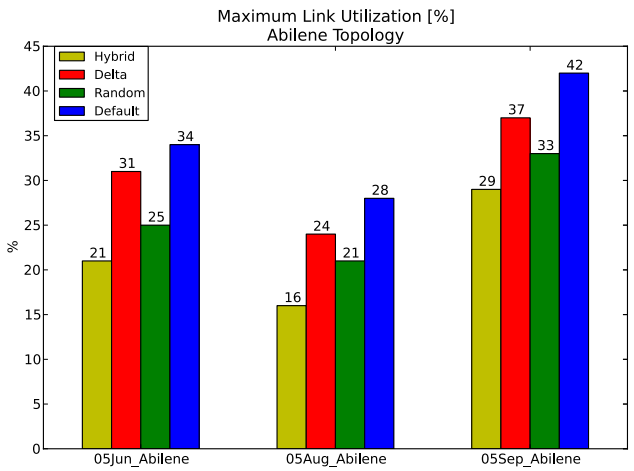


Fig. 11 MLU versus approach, abilene topology

node is turned off with its links, then its power consumption is zero. Neglecting the traffic is based on checking if the amount of link utilization is smaller than a lower limit (we selected 5% as lower limit). This has a tremendous effect on the total power of the network.

Across all traffic simulations, we keep track of the MLU, while turning off those nodes that have low link utilization. In all three approaches, and after each execution of our MOO, we keep track of the *top* node; the node that has the highest aggregate link capacity among the nodes that have been selected to be turned off so far. This *top* node yields the highest traffic capacity considering all links that are connected to it. If the MLU increases above a specific thresholdα, then we turn this *top* node back on. In order not to cause high MLU and not to cause congestion, we turn on *top* node which possesses the highest incoming and outgoing traffic capacity that can resolve the extra capacity required in the network [6].

The performance of the three sets of traffic patterns using the LF TE scheme is demonstrated in Table 3. It shows the amount of savings in PC and MLU for each traffic pattern. All three approaches are compared to the results obtained when we use the default link weights. In the case of hybrid approach, the saving for the September traffic pattern can reach 35.24% in PC while the MLU is also reduced by 30.95%. We notice that delta weight approach has the lowest improvement in both PC and MLU when compared with other two approaches. Random approach searches solutions in a larger search space than the delta approach and is less prone to being stuck in a locally optimal region. Therefore, the solutions found by random approach give much better savings in MLU and PC than the delta approach. Eventually combining both approaches of random and delta into the hybrid approach gives the highest saving in both PC and MLU. Compared to the delta approach, the hybrid approach starts with a solution found by the random approach, which is much better than the default initial solution used by the standalone delta weight approach. It then fine tunes the link weights using the delta weight approach to continue searching for better solutions. It has the good features of both the other two approaches and returns the best solution among all three approaches.

5.2.3 Polska network topology results

The second network topology that we used for testing our research approach is Polska network topology. We created three flavors of traffic which were deployed using the LF TE scheme. The three traffic flow types vary the link load utilization by using low, medium or high traffic load. Simulation validation for those three traffic

Table 3 PC and MLU savings for three sets of traffic, abilene topology

Abilene traffic period	PC Savings versus default			MLU Savings versus default		
	Random (%)	Delta (%)	Hybrid (%)	Random (%)	Delta (%)	Hybrid (%)
June	25.68	16.51	33.09	26.47	8.82	38.24
August	20.00	16.68	32.45	25.00	14.29	42.86
September	29.75	15.03	35.24	21.43	11.90	30.95

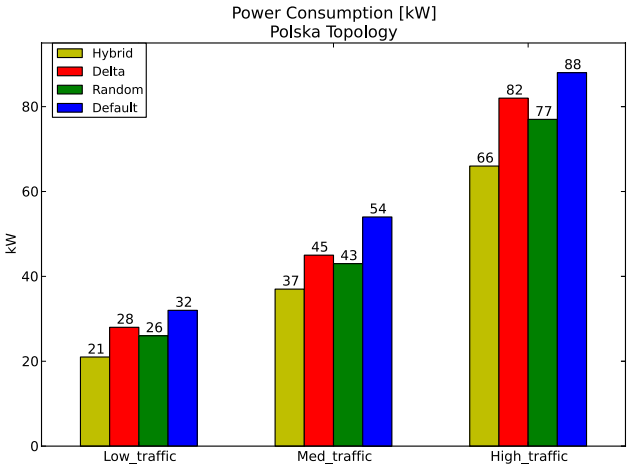


Fig. 12 PC versus approach, polska topology

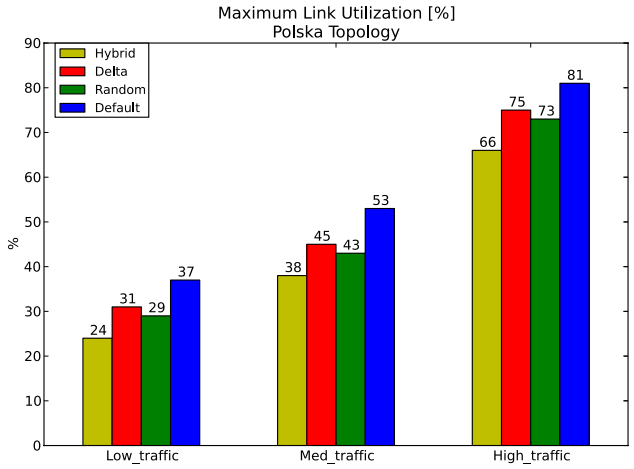


Fig. 13 MLU versus approach, polska topology

profiles includes various types of traffic patterns, such as Poisson, Periodic and Flat flows.

The default baseline approach was chosen to have equal link weights for all links, and we selected a value of 32500, which is almost half the range allowed for any link weight as described in [37]. The three other approaches are the random, delta and hybrid approaches that we discussed in Sect. 4. In Fig. 12, we show the simulation results of the PC for the four scenarios. We can see that the PC is much smaller in the case of the low traffic scenario. The hybrid approach gives the highest power savings, especially when we have a high traffic scenario.

Figure 13 describes the MLU obtained for the four different approaches that we simulated. The hybrid approach gives the lowest MLU value. Table 4 summarizes the combined results from all simulations conducted. We used three traffic flows that mimic various link load scenarios from low to high network link loads. The range of link loads that we selected was based on altering the traffic flows using MGEN so that the link load utilization is between 30 and 90%. We measured link utilization such as low traffic scenario can reach up to 35%, while medium scenario matches 55% and high link load utilization should be around 85%. Using MGEN, we had chosen about 15, 25 and 30 traffic flows to be used for low, medium and high scenarios, respectively.

We can see from Table 4 that as we increase the traffic level from low to high, the savings of MLU for the hybrid approach is decreased from 35.14 down to 18.52%. For the power savings, we do get similar savings from 34.38 down to 25%. When the network traffic load is low, most links and nodes are lightly used and our approaches are able to use only some of them to carry the traffic and put the remaining ones into sleep to save PC and achieve acceptable MLU. In the high traffic level scenario, almost all nodes and links are needed to carry the network traffic and there is less room for our approaches to find the nodes and links to be turned off. However, even in high traffic level scenarios, our approaches were still able to achieve decent improvement compared to the default baseline approach. Polska topology results show that the hybrid approach is more effective than the random and delta weight approaches.

6 Conclusions and future research

We proposed a novel genetic algorithm based Multi-Objective optimization scheme for reducing PC while maintaining acceptable MLU in wired networks. We presented three approaches to search for the best link weights configuration for the current network traffic and topology. The first approach uses a random method

Table 4 PC and MLU savings for three sets of traffic, polska topology

Polska traffic flow	PC Savings versus default			MLU Savings versus default		
	Random (%)	Delta (%)	Hybrid (%)	Random (%)	Delta (%)	Hybrid (%)
Low	18.75	12.50	34.38	21.62	16.22	35.14
Med.	20.37	16.67	31.48	18.87	15.09	28.30
High	12.50	6.82	25.00	9.88	7.41	18.52

for selecting the initial set of link weights, while the second approach uses $\Delta w$for varying the link weights, and in the third approach, a hybrid method is used by combining both the first and second approaches. All three approaches use MOGA for finding new solutions that are applied to the network. GA solutions are obtained by using DEAP Python software and the best solution is tracked in each GA iteration. Two network topologies of Abilene and Polska were used for testing the three approaches. Network traffic dataset for Abilene was selected from three different dates based on [41], while for Polska topology we created three different synthetic traffic loads. Best results were achieved by the hybrid approach which shows that for Abilene network topology, we can reduce PC from 32.45 to 35.24%, while reducing MLU in average from 30.95 to 42.86% for specific traffic pattern. For Polska network topology, our hybrid approach showed a reduction in average from 25 to 34.38% in PC and MLU savings in the range of 18.52–35.14%. The results of this research allow various recommendations to be considered by the network operators in order to save PC without increasing link utilization of the network. The average runtime of our algorithm takes a few minutes for every iteration, after waiting for a few seconds until the next iteration. This hybrid approach can fit other network topologies, and can be easily adapted to run in parallel with multiple CPUs, and thus reduce the runtime.

In summary, we advanced the current state of energy aware networks by the effort presented in this research. We demonstrated that a multi-objective optimization module using MOGA can be applied to any wired network to help optimize the network configuration such as link weights. The methods presented were able to find the optimal link weights configuration to reduce PC while not affecting MLU and thus enabling network operators to save network energy without affecting their network load balancing. Future research would consider merging our approach with optimization of traffic pattern analysis and measurement. Our simulation provided a consistent and repeatable environment in order to compare different approaches using the same datasets of traffic patterns. The best way to show how a similar future approach can be effective in saving energy without affecting MLU is to run it autonomously on real online traffic data. Continuous analysis of online traffic is required due to the prevailing changing nature of Internet traffic. A second recommendation is to expand this approach for supporting larger network topologies. Larger network topologies exhibit a different challenge for converging the network flow after changing the link weights. Therefore, further study into the algorithm's parameters, mutation, crossover, and Δwvalues is inevitable and may yield a superior approach for handling larger networks that comprise many links and nodes.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Vallet J, Brun O (2014) Online OSPF weights optimization in IP networks. Comput Netw 60:1–12 ISSN 1389-1286
2. Addis B, Capone A, Carello G, Gianoli LG, Sanso B (2012) Energy-aware multiperiod traffic engineering with flow-based routing. In: 2012 IEEE international conference on communications (ICC), pp 5957–5961. IEEE. ISBN 1457720523
3. Chiaraviglio L, Mellia M, Neri F (2012) Minimizing ISP network energy cost: formulation and solutions. IEEE/ACM Trans Netw 20(2):463–476 ISSN 1063-6692
4. Tharwat A, Houssein EH, Ahmed MM, Hassanien AE, Gabel T (2018) MOGOA algorithm for constrained and unconstrained multi-objective optimization problems. Appl Intell 48(8):2268–2283
5. Cui Y, Geng Z, Zhu Q, Han Y (2017) Multi-objective optimization methods and application in energy saving. Energy 125:681–704
6. Francois F, Wang N, Moessner K, Georgoulas S, Xu K (2014) On IGP link weight optimization for joint energy efficiency and load balancing improvement. Comput Commun 50:130–141 ISSN 0140-3664
7. Andrae ASG, Edler T (2015) On global electricity usage of communication technology: trends to 2030. Challenges 6(1):117–157
8. Bianzino AP, Chaudet C, Larroca F, Rossi D, Rougier J-L (2010) Energy-aware routing: a reality check. In: GLOBECOM Workshops (GC Wkshps), pp 1422–1427. IEEE. ISBN 142448863X
9. 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 Commun Surv Tutor 13(2):223–244 ISSN 1553-877X
10. Bianzino AP, Chaudet C, Rossi D, Rougier J (2012) A survey of green networking research. IEEE Commun Surv Tutor 14(1):3–20 ISSN 1553-877X

11. Morley J, Widdicks K, Hazas M (2018) Digitalisation, energy and data demand: the impact of Internet traffic on overall and peak electricity consumption. Energy Res Soc Sci 38:128–137
12. Addis B, Capone A, Carello G, Gianoli LG, Sansò B (2016) Energy management in communication networks: a journey through modelling and optimization glasses. Comput Commun. ISSN 0140-3664
13. Van Heddeghem W, Lambert S, Lannoo B, Colle D, Pickavet M, Demeester P (2014) Trends in worldwide ICT electricity consumption from 2007 to 2012. Comput Commun 50:64–76
14. Vasić N, Bhurat P, Novaković D, Canini M, Shekhar S, Kostić D (2011) Identifying and using energy-critical paths. In: Proceedings of the seventh conference on emerging networking experiments and technologies, p 18. ACM. ISBN 1450310419
15. Zhang M, Yi C, Liu B, Zhang B (2010) GreenTE: power-aware traffic engineering. In: The 18th IEEE international conference on network protocols, pp 21–30. IEEE
16. Fisher W, Suchara M, Rexford J (2010) Greening backbone networks: reducing energy consumption by shutting off cables in bundled links. In: Proceedings of the first ACM SIGCOMM workshop on Green networking, pp 29–34. ACM. ISBN 1450301967
17. Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N (2010) ElasticTree: saving energy in data center networks. In: NSDI, vol 10, pp 249–264
18. Dabaghi F, Movahedi Z, Langar R (2017) A survey on green routing protocols using sleep-scheduling in wired networks. J Netw Comput Appl 77:106–122
19. Bouras I, Figueiredo R, Poss M, Zhou F (2020) Minimizing energy and link utilization in ISP backbone networks with multi-path routing: a bi-level approach. Optim Lett 14(1):209–227
20. Idzikowski F, Chiaraviglio L, Cianfrani A, Vizcaino JL, Polverini M, Ye Y (2016) A survey on energy-aware design and operation of core networks. IEEE Commun Surv Tutor 18(2):1453–1499 ISSN 1553-877X
21. Majumder S, Kar MB, Kar S, Pal T (2020) Uncertain programming models for multi-objective shortest path problem with uncertain parameters. Soft Comput 24(12):8975–8996
22. Lei J, Deng S, Lu Z, He Y, Gao X (2020) Energy-saving traffic scheduling in backbone networks with software-defined networks. Clust Comput 6:1–14
23. Antonio C, Vincenzo E, Marco L, Marco M, Enrico V (2010) An energy saving routing algorithm for a green OSPF protocol. In: INFOCOM IEEE conference on computer communications workshops 2010, pp 1–5. IEEE. ISBN 142446739X
24. Padhye N, Bhardawaj P, Deb K (2013) Improving differential evolution through a unified approach. J Global Optim 55(4):771–799
25. Deb K (2011) Multi-objective optimisation using evolutionary algorithms: an introduction. In: Multi-objective evolutionary optimisation for product design and manufacturing, pp 3–34. Springer
26. Ahmadi MH, Mohammadi AH, Dehghani S, Barranco-Jimenez MA (2013) Multi-objective thermodynamic-based optimization of output power of Solar Dish-Stirling engine by implementing an evolutionary algorithm. Energy Conver Manag 75:438–445
27. Aziz MS, Zarei S, Mansoorizadeh M, Nassiri M (2018) Toward energy-aware traffic engineering in intra-domain IP networks using heuristic and meta-heuristics approaches. Inf Syst Telecommun 5:95
28. Samadi R, Nassiri M, Mansoorizadeh M (2020) Energy-aware traffic engineering in IP networks using non-dominated sorting genetic II algorithm. Int J Adv Intell Paadigms 16(1):75–87
29. Celenlioglu MR, Goger SB, Mantar HA (2014) An SDN-based energy-aware routing model for intra-domain networks. In: 2014 22nd international conference on software, telecommunications and computer networks (SoftCOM), pp 61–66. IEEE
30. Assefa BG, Özkasap Ö (2019) A survey of energy efficiency in SDN: software-based methods and optimization models. J Netw Comput Appl 137:127–143
31. Maaloul R, Taktak R, Chaari L, Cousin B (2018) Energy-aware routing in carrier-grade ethernet using sdn approach. IEEE Trans Green Commun Netw 2(3):844–858
32. Yazbek H, Liu P (2019) Adaptive strategies of multi-objective optimization for greener networks. In: 2019 SoutheastCon, pp 1–8. IEEE
33. Yazbek H (2019) Adaptive strategies of multi-objective optimization for greener networks. Ph.D. thesis, Nova Southeastern University, Fort Lauderdale, Florida 33314-7795, USA, 12. Retrieved from NSUWorks, College of Computing and Engineering
34. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lect Notes Comput Sci 1917:849–858 ISSN 0302-9743
35. Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248
36. Ericsson M, Resende MGC (2002) A genetic algorithm for the weight setting problem in OSPF routing. J Comb Optim 6(3):299–333 ISSN 1382-6905
37. Fortz B, Thorup M (2000) Internet traffic engineering by optimizing OSPF weights. In: Proceedings of INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. IEEE, vol 2, pp 519–528. IEEE. ISBN 0780358805
38. Amaldi E, Capone A, Gianoli LG (2013) Energy-aware IP traffic engineering with shortest path routing. Comput Netw 57(6):1503–1517 ISSN 1389-1286
39. Ahrenholz J, Danilov C, Henderson TR, Kim JH (2008) CORE: a real-time network emulator. In: Military communications conference, 2008. MILCOM 2008. IEEE, pp 1–7. IEEE. ISBN 1424426766
40. De Rainville FF-A, Gardner M-A, Parizeau M, Gagné C (2012) DEAP: A python framework for evolutionary algorithms. In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation, pp 85–92. ACM. ISBN 1450311784
41. Orlowski S, Wessäly R, Pióro M, Tomaszewski A (2010) SNDlib 1.0 survivable network design library. Networks 55(3):276–286 ISSN 1097-0037
42. Tune P, Roughan M, Haddadi H, Bonaventure O (2013) Internet traffic matrices: A primer. Rec Adv Netw 1:1–56
43. Gianoli LG (2014) Energy-aware traffic engineering for wired IP networks. Departement De G Enie' Electrique' Ecole Polytechnique De Montreal
44. Moulierac J, Phan K (2014) Optimizing IGP link weights for energy-efficiency in a changing world. Comput Commun 6:19