HEURISTIC ALGORITHMS FOR JOINT OPTIMIZATION OF UNICAST AND ANYCAST TRAFFIC IN ELASTIC OPTICAL NETWORK–BASED LARGE–SCALE COMPUTING SYSTEMS

MARCIN MARKOWSKI [a]

[a]Department of Systems and Computer Networks, Faculty of Electronics
Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: `marcin.markowski@pwr.edu.pl`

In recent years elastic optical networks have been perceived as a prospective choice for future optical networks due to better adjustment and utilization of optical resources than is the case with traditional wavelength division multiplexing networks. In the paper we investigate the elastic architecture as the communication network for distributed data centers. We address the problems of optimization of routing and spectrum assignment for large-scale computing systems based on an elastic optical architecture; particularly, we concentrate on anycast user to data center traffic optimization. We assume that computational resources of data centers are limited. For this offline problems we formulate the integer linear programming model and propose a few heuristics, including a meta-heuristic algorithm based on a tabu search method. We report computational results, presenting the quality of approximate solutions and efficiency of the proposed heuristics, and we also analyze and compare some data center allocation scenarios.

Keywords: elastic optical networks, optimization, distributed data center, algorithm, tabu search.

1. Introduction

Bandwidth consuming network services, like IP video and business applications, entail an incessant increase in the network traffic volume and the amount of data sent. The substantial impact of traffic growth is also a result of the cloud computing model, more and more popular and expanding during recent years. User data and computational resources are no longer localized on user or company premises, but are available in the cloud. Thus it is necessary to use the global computer network for access to data or business applications. According to the Cisco forecast (Cisco, 2014), average Internet traffic will increase 2.8-fold, and busy-hour Internet traffic even 3.4-fold over the years 2013–2018, finally exceeding 1 Pbit per second. Increasing user demands will force network providers and operators to apply advanced and more efficient methods of network bandwidth management, including prospecting of new technologies and algorithms which will guarantee better utilization of the given network infrastructure.

In this context, a very attractive and promising solution is the elastic optical network (also called the SLICE—spectrum-sliced elastic optical path network) architecture. This technology enables the division of a spectrum into slots (or slices or subcarriers) of 6.25 GHz width, and building optical channels consisting of an even number of adjacent slots (ITU-T, 2012).

The concept of the SLICE network is based on usage of bandwidth-variable transponders as terminating elements of a lightpath and wavelength cross connects (WXCs) for transit optical switching (Jinno *et al.*, 2009). Transponders are responsible for the proper converting of the data signal at the beginning of the lightpath and retrieving it at the end. Depending on the length of the optical path (and signal regeneration possibilities), the most convenient modulation format is applied, e.g., BPSK, QPSK, 8QAM. The necessary bitrate is adjusted by allocating an appropriate amount of spectral resources (number of subcarriers). Optical channels in elastic optical networks (EONs) must be built using a continuous sequence of subcarriers in order to guarantee spectrum contiguity. WXCs are used in transit nodes, their task being to switch the optical signal between input and output ports in order to establish an end-to-end lightpath. The rule of spectrum continuity must be observed—the same

spectrum range must be used along the whole lightpath (Chatterjee *et al*., 2015). The SLICE architecture makes it possible to elastically fit the optical channel bandwidth for user demands, and to create optical channels of a relatively small (sub-wavelength accommodation) as well as an exceeding 100 Gb/s (super-wavelength accommodation) bandwidth. Features and architecture of elastic optical networks were widely described by Jinno *et al*. (2009) and Layec *et al*. (2013).

Together with migration to the cloud computing model, the pressure on scalability and reliability of cloud-available services is growing (Yoo *et al*., 2013). Distributed and cloud computing technologies are ready to satisfy those demands, services are rendered by numerous servers or data centers, geographically distributed and cooperating as one logical entity. The ability of choosing one or a few of them for particular user demand gives the cloud provider the opportunity to select the optimal data center (or set of data centers) for demand processing. The decision may be based on network latency, availability of computing resources, cost (Wubin *et al*., 2013), etc. Delegating demand to many data centers ensures scalability and increases clients' quality of experience by reducing processing time. Working in the cloud environment entails, in its matter, taking advantage of anycast traffic. Moreover, two types of anycast traffic may be distinguished while analyzing distributed computing systems: classic anycast transmission (to one data center) or multi-destination anycast (in the case when demand is split into sub-demands and each sub-demand is sent to another data center in the anycast group).

There are many computational power consuming applications in different fields of science, e.g., for satellite imagery processing (Warren, 2015), weather forecasting, biomedical processing (Cierniak, 2014) or big-data processing (Wang *et al*., 2016). Efficient processing, or real-time processing, of huge volumes of data more and more frequently requires utilization of distributed computing resources since one machine or data center does not offer a satisfactory processing power. In such a processing model, huge amounts of data (e.g., raw bitmap satellite imagery) are sent for processing to data/computational centers. In many cases the processing time is proportional to the volume of incoming data. When processing is completed, results are to be sent backward to the client; usually the volume of results is significantly smaller (e.g., vector maps made on the basis of bitmaps or conclusions derived from big data) than raw data sent to the data center (Warren, 2015). For faster task completion, or in the case of tasks which exceed the power of a single data center, raw data may be divided into portions and few data centers may be selected for processing. In the paper we focus on such a real-time data processing model: we assume a high data transfer rate from a client to the data centers (a lot

of raw data), a possibility of dividing tasks into subtasks and processing at many data centers (and generating multi-destination anycast traffic), the required processing power proportional to the incoming data stream, and relatively small data rates of results sent from the data center to the client side.

We consider distributed large-scale computing systems consisting of data centers and client systems connected to the nodes of an elastic optical network. Data centers offer computational services, and the amount of computational (processing) resources available at each data center is limited. Client systems, localized across the network, utilize computational power of data centers in order to complete computational tasks. Each task may be executed at one data center or may be divided into portions and executed at many data centers. Then for each portion of a computational task we consider anycast or multi-destination anycast traffic from the client system to the data center. Apart from user–data center traffic, we assume that direct unicast data transmission between client systems must be accomplished at the same time in the network. The optimization problem consists in task distribution over data centers and allocation of the routes and spectrum for anycast and unicast traffic in order to minimize the maximal or average usage of the spectrum in the optical network. For each data transmission demand the optical channel in the EON must be assigned, which means that the route over the network and the spectrum (sequence of slices) along the chosen route must be selected. In the network there are many possible routes (paths) between each pair on nodes. We assume that for each node pair the set of alternative candidate paths is given, and one path must be chosen for each data transmission.

The main novelty and contribution of this study is the formulation of anycast user–data center and unicast user–user traffic optimization problem in elastic optical network-based large-scale computing systems with constraints on the processing power of data centers and the possibility of dividing computational tasks for simultaneous processing at many data centers. We introduce an integer linear programming (ILP) formulation of the problem considered for two objective functions: average and maximal spectrum usage. For both the objectives we propose an improved tabu search method, offering a faster discovery of local optima. We report results of extensive computational experiments, discussing the quality of tabu search solutions and properties of the problem considered. Finally, a case study of different data center allocation scenarios is reported.

The paper is organized as follows. In Section 2 we present a review of EON optimization problems and solutions considered in the literature. The network model and ILP formulation are described and discussed in Section 3. In Section 4 we propose heuristic

greedy algorithms, and in Section 5 a tabu search-based meta-heuristic. The experimental setup, the tuning process of the tabu algorithm and results of experiments are described in Section 6. Section 7 concludes the paper.

2. Related works

Route assignment in elastic optical networks is strictly connected with the spectrum reservation necessity along the lightpath. The most important problem considered for the EON is routing and spectrum assignment (RSA) (Talebi *et al.*, 2014; Klinkowski and Walkowiak, 2011), consisting in simultaneous route selection and reservation of a continuous range of the spectrum along the path. The maximal range of an optical signal is limited and strictly depends on the chosen modulation level (Jinno *et al.*, 2009; Layec *et al.*, 2013). Higher modulation levels enable us to increase the throughput of the optical channel, but they decrease the maximal range of the signal. Then, in RSA problems the possibility of using different modulation levels is often considered (Goścień *et al.*, 2014; Talebi *et al.*, 2014). The range of the optical signal may be extended using optical regenerators. In the work of Fallahpour *et al.* (2014) or Aibin and Walkowiak (2015) the regenerator placement problem is considered.

Most of the research concentrates on the off-line RSA problem, when the traffic demands are of long-term nature and are known in advance. The most important optimization criteria considered in off-line RSA indicate the minimization of spectrum usage (Walkowiak and Klinkowski, 2013): the goal is to minimize the total (or average) number of occupied slices. Other formulations take into account other indicators as criteria or optimization constraints: spectrum fragmentation (Talebi *et al.*, 2014), the number and placement of regenerators (Aibin and Walkowiak, 2015), survivability demands (Walkowiak *et al.*, 2014b; Goścień *et al.*, 2015), the total number (or cost) of optical transceivers installed in the network (Tornatore *et al.*, 2015), etc. Optimization problems are formulated for unicast traffic (Velasco *et al.*, 2012; Klinkowski and Walkowiak, 2011), anycast (Zhang and Zhu, 2014), multicast (Liu *et al.*, 2013) or combined (Walkowiak and Klinkowski, 2013).

Off-line RSA is an NP-complete problem (Chatterjee *et al.*, 2015; Christodoulopoulos *et al.*, 2011). In the literature, ILP models may be found (Klinkowski and Walkowiak, 2011; Velasco *et al.*, 2012; Walkowiak and Klinkowski, 2013). They are useful for small network topologies. For bigger problem complexities, heuristic and meta-heuristic approaches are considered (Klinkowski and Walkowiak, 2011; Walkowiak and Klinkowski, 2013; Walkowiak *et al.*, 2014a). Wang *et al.* (2011) prove the NP-hardness of the RSA problem, propose an ILP model for solving it for small network topologies, analyze lower and upper bounds of spectrum

utilization and present two heuristic algorithms for efficiently solving the RSA problem for bigger networks.

In the dynamic (on-line) RSA problem, connection requests appear dynamically and must be immediately allocated in the network (Walkowiak *et al.*, 2014a). The most important optimization criterion is the blocking probability (Zhang and Zhu, 2014), but other criteria are also taken into account. Exact algorithms are not sufficient for online problems, as decisions must be taken in milliseconds; then heuristic algorithms are proposed. Most RSA approaches assume utilization of a single path for each demand in the EON. Zhu *et al.* (2013) notice that for online provisioning it is often impossible to serve large requests with single-path routing, while multi-path routing allows such allocation. They proposed several algorithms and verified that the use of multipath routing can significantly decrease the blocking probability in online RSA optimization.

The problem of applying EONs to data center traffic optimization was considered by Yoo and Wen (2012). Zhang and Zhu (2014) proposed several heuristic algorithms for solving a dynamic RSA problem in the EON for inter-data center traffic; moreover, some data center resource limitations were taken into account. A tabu-search algorithm for the RSA problem in EON-based distributed computing systems was proposed by Markowski (2015). Fang *et al.* (2015) present an MILP model and solution algorithms for defragmentation of different data center resources (like computing power, memory, disc space). At first, demands are allocated to data centers according to free resources and fragmentation avoidance, after that the optical path and spectrum are assigned for traffic demands. Some proposals for using EONs for clouds were presented by Goścień *et al.* (2014) and Yoo *et al.* (2013).

In this paper we formulate a novel spectrum usage optimization problem in an EON-based large-scale computing system, taking into account constraints put on the processing power of data centers. We propose an ILP model and a tabu-search meta-heuristic. To the best of our knowledge, the so formulated problem with a processing power constraint has never been studied before.

3. Network model

The elastic optical network is modeled as a directed graph with V nodes labeled with v ($v = 1, 2, \ldots, V$) and E links labeled with e ($e = 1, 2, \ldots, E$). R data centers are connected to the nodes of the network. For each data center, the node and the computational power are predetermined. Computational tasks are defined with the number of the client node, the volume of traffic demand that must be uploaded to the data center (*anycast upstream demand*), the number of instructions required process the uploaded demand, and the volume of the

results which must be downloaded from the data center to the client system (*anycast downstream demand*). Each computational task can be executed in one data center or may be divided into subtasks and sent for processing to a few data centers. In the latter case, the computational task generates data flows to a few chosen date centers, as well as a few feedback flows. Those data flows will be called 'streams' in our further discussion: *anycast upstream streams* from the client system to data centers and *anycast downstream streams*—the feedback to the client. Unicast traffic between client systems is given with source and destination nodes, and the volume of demand.

Traffic demands are labeled with d; D is the total number of demands, and A is the number of anycast demands. Here $d = 1, 2, \ldots, A$ are anycast upstream demands, $d = A + 1, A + 2, \ldots, 2A$ are anycast downstream demands and $d = 2A + 1, 2A + 2, \ldots, D$ are unicast demands. We assume that the volumes of anycast downstream demands are significantly lower than those of anycast upstream demands, as described in Section 1. Upstream and downstream the demands realizing the same anycast request are called associated. For each of them, $\tau(d)$ indicates the demand associated with demand d. For each data center used for demand $d = 1, 2, \ldots, 2A$, the upstream and downstream streams must be routed in the network. The number of demand streams is limited by the number of data centers. The volume of each demand is denoted by h_d and given in Gbps. We assume that the volume of the downstream stream is proportional to that of the upstream stream.

We assume that at most one data center is located in each network node (or many data centers connected to one node are treated as one and their processing powers are summed up). Data centers have limited processing power c_v (in GIPS—giga instructions per second). The number of instructions required to process demand d in a data center is given by λ_d, where λ_d denotes the number of instructions required to process the data of demand d transmitted with a streaming rate of 1 Gbps and is given in GIPS/Gbps. We assume that computational tasks with the same λ generated by all users connected to the same network node are aggregated and considered in the optimization problem to be one computational task.

A simple example is presented in Fig. 1. The network consists of five nodes and two data centers are connected to the nodes. Their processing power is $c_1 = 4000$ GIPS, $c_3 = 3000$ GIPS. Users at Node 2 need to complete a computational task (Task 1), which generates anycast upstream demand with volume of 150 Gbps (Demand 1), anycast downstream demand with the volume of 3 Gbps (Demand 3), and $\lambda = 30$ GIPS/Gbps. Similarly, users at Node 5 generate Task 2, with anycast upstream demand of 100 Gbps (Demand 2), anycast downstream demand of 5 Gbps (Demand 4), and $\lambda = 10$ GIPS/Gbps. At the same time unicast demand (Demand 5) from users at Node 4
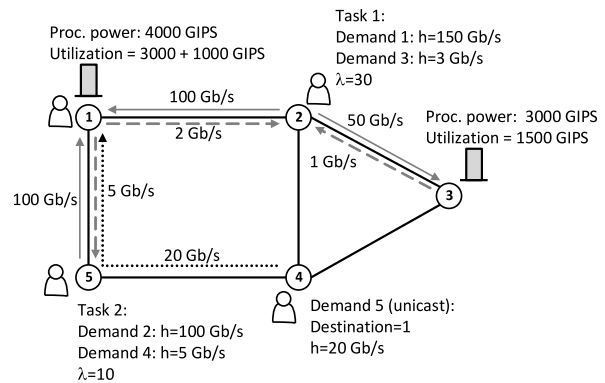

Fig. 1. Example of task processing.

to users at Node 1, with the volume of 20 Gbps must be routed in the network. An exemplifying solution for the scenario is the following. Task 1 is being processed at two data centers. Then the upstream and downstream traffic is divided into two upstream streams (100 Gbps to the data center at Node 1 and 50 Gbps to data center at Node 3) and two downstream streams (with volumes proportional to the associated upstream streams). Task 2 is processed at one data center. Utilization of the data center at node 1 is equal to 100 Gbps × 30 GIPS/Gbps (part of Task 1) + 100 Gbps × 10 GIPS/Gbps (Task 2) = 4000 GIPS (100% utilization), whereas utilization of the data center at Node 3 is equal to 50 Gbps × 30 GIPS/Gbps (part of Task 1) = 1500 GIPS.

We assume k candidate paths for each pair of nodes of the network. For each demand d there is a set of P_d candidate routing paths, indexed by $p = 1, 2, \ldots, P_d$. For each unicast demand we have a set of k candidate paths, out of which exactly one must be used. For each anycast demand the number of candidate paths is equal to Rk or $(R - 1)k$ if one of the data centers is located in the node which is the source of upstream demand. For each anycast demand, R (or $R - 1$) paths may be chosen, each of them routing to another data center. For the above-considered example, the routes chosen for all traffic demands (or streams) are presented in Fig. 1.

The available spectrum resources in each optical link are divided into a set of S frequency slices. Optical channels can be built with an even number of adjacent slices. The possible sizes of channels are denoted by $m = 2, 4, \ldots, M$ slices. On each optical link we may build a different number of channels, depending on their sizes. For anycast demands, the size of channels must be chosen according to the volume of the demand streams, since each anycast demand may use many data centers. Channels of size m used for realizing anycast demands are indexed with c, where $c = 1, 2, \ldots, C_m$. For unicast demand the required size of a channel for each path is known, since each unicast demand is carried over a

single path. Candidate channels for unicast demand d on path p are indexed with g, where $g = 1, 2, \ldots, G_{dp}$. The maximal capacity of optical channels depends on the length of the lightpath and the number of slices.

We consider an elastic optical network consistent with the ITU frequency grid definition, i.e., ITU-T Recommendation G.694.1 (Ed. 2.0), Spectral grids for WDM applications: DWDM frequency grid (ITU-T, 2012). According to the definition, the spectrum has to be allocated symmetrically around a central frequency and the frequency slice width is equal to $\Delta = 6.25$ GHz. We apply the half distance law and assume that the 16QAM, 8QAM, QPSK and BPSK modulation formats can be used with transmission up to 375 km, 750 km, 1500 km and above 1500 km, respectively (Christodoulopoulos *et al.*, 2011; Walkowiak and Klinkowski, 2013). Moreover, we assume that transponders operate with polarization division multiplexing, i.e., the spectral efficiency is doubled and equal to $2n_p$ [bit/s/Hz], where n_p is the modulation level for lightpath p, equal to 4,3,2 and 1, respectively, for modulation formats BPSK, QPSK, 8QAM and 16QAM. The maximum capacity β_{dpm} of lightpath p of demand d with the width of m slices is equal to $2n_p m\Delta$, [Gbps].

We consider two objective functions representing spectrum usage in EON. One of them, the ToS (*total spectrum*), represents the number of frequency slices which are used on any network link. The value of the ToS is an answer to the question: What amount of the spectrum is required for successful allocation of all traffic demands? The ToS is usually equal to the largest index of the allocated slices, but sometimes this might not be true due to spectrum fragmentation. The other objective, the AvS (*average spectrum*), represents the average width of the spectrum (number of slices) used on network links. Minimization of the AvS allows irregular utilization of network links, while in order to minimize the ToS the utilization of links should be as uniform as possible.

3.1. Optimization model.
In order to formulate the ILP model, we use the notation proposed by Walkowiak and Klinkowski (2013), Klinkowski and Walkowiak (2011) as well as Markowski (2015).

Indices
$s \quad = 1, 2, \ldots, S$, slices
$m \quad = 2, 4, \ldots, M$, sizes of candidate channels (in number of slices)
$c \quad = 1, 2, \ldots, C_m$, candidate channels of size m (for anycast)
$g \quad = 1, 2, \ldots, G_{dp}$, candidate channels for unicast demand d on path p

$v \quad = 1, 2, \ldots, V$, network nodes
$d \quad = 1, 2, \ldots, D$, demands (both anycast and unicast)
$d \quad = 1, 2, \ldots, A$, upstream anycast demands
$d \quad = A + 1, 2, \ldots, 2A$, downstream feedbacks
$d \quad = 2A + 1, 2, \ldots, D$, unicast demands
$p \quad = 1, 2, \ldots, P_d$, candidate paths for flows realizing demand d; path p connects the client node and the data center node
$e \quad = 1, 2, \ldots, E$, network links

Constants
$\delta_{edp} \quad = 1$, if link e belongs to path p realizing demand d; 0, otherwise
$h_d \quad$ volume of demand d (Gbps)
$\gamma_{mcs} \quad = 1$, if channel c of size m uses slice s; 0, otherwise
$\gamma_{dpgs} \quad = 1$, if channel g associated with unicast demand d on path p uses slice s; 0, otherwise
$\beta_{dpm} \quad$ maximum capacity of path p of anycast demand d with width of m slices (Gbps)
$t_{dpv} \quad = 1$, if node v is the destination node of path p for demand d; 0, otherwise
$u_{dpv} \quad = 1$, if node v is the source node of path p for demand d; 0, otherwise
$c_v \quad$ processing power of server at node v (GIPS)
$\lambda_d \quad$ number of instructions required to process data of demand d (anycast upload) transmitted with the streaming rate of 1 Gbps (GIPS/Gbps)
$\tau(d) \quad$ index of anycast demand associated with demand d; if d is downstream demand, then $\tau(d)$ must be an upstream connection and vice versa

Variables
$x_{dpmc} \quad = 1$, if channel c of size m on candidate path p is used to realize anycast demand d; 0, otherwise (binary)
$x_{dpv} \quad$ flow of anycast demand d, assigned to path p, sent to server located at node v, given in Gbps (continuous)
$z_{dpg} \quad = 1$, if channel g on candidate path p is used to realize unicast demand d; 0, otherwise (binary)
$y_{es} \quad = 1$, if slice s is occupied on link e; 0, otherwise (binary)
$y_s \quad = 1$, if slice s is occupied on any link in the network; 0, otherwise (binary)
$q_e \quad$ largest index of allocated slices in link e (integer)

In the optimization model we use six sets of decision variables. The binary variable x_{dpmc} denotes the selection of the path, size and number of a channel for demand d. The continuous variable x_{dpv} denotes the volume of the anycast demand stream, being the part of demand d which

is sent along path p to data centers located at node v. The binary variable z_{dpg} denotes the selection of the path and channel for unicast demand d. The binary variable y_{es} indicates whether the slice s on link e is occupied. The binary variable y_s indicates whether the slice s is occupied on any network links. Finally, the integer variable q_e denotes the largest index of allocated slices on link e.

3.2. Total spectrum optimization: Problem formulation.

The goal of total spectrum optimization (ToSO) is to minimize the number of frequency slots used anywhere in the EON. Slice s is 'used' when it is occupied in any of the network links. The number of slots used is the objective in the ToS optimization problem:

objective

$$\text{minimize } ToS = \sum_{s=1,\ldots,S} y_s \qquad (1)$$

subject to

$$x_{dpv} \le \sum_{m=2,4,\ldots,M} \sum_{c=1,\ldots,C} t_{dpv}\beta_{dpm}x_{dpmc},$$
$$d = 1,\ldots,2A, \quad p = 1,\ldots,P(d), \quad v = 1,\ldots,V, \quad (2)$$

$$\sum_{p=1,\ldots,P} \sum_{v=1,\ldots,V} x_{dpv} = h_d, \quad d = 1,\ldots,D, \qquad (3)$$

$$\sum_{d=1,\ldots,2A} \sum_{p=1,\ldots,P} \sum_{m=2,4,\ldots,M} \sum_{c=1,\ldots,C} \gamma_{mcs}\delta_{edp}x_{dpmc}$$
$$+ \sum_{d=2A+1,\ldots,D} \sum_{p=1,\ldots,P} \sum_{g=1,\ldots,G} \gamma_{dpgs}\delta_{edp}z_{dpg} \le y_{es},$$
$$e = 1,\ldots,E, \quad s = 1,\ldots,S, \quad (4)$$

$$\sum_{e=1,\ldots,E} y_{es} \le Ey_s, \quad s = 1,\ldots,S, \qquad (5)$$

$$\sum_{d=1,\ldots,A} \sum_{p=1,\ldots,P} \lambda_d x_{dpv} \le c_v, \quad v = 1,\ldots,V, \qquad (6)$$

$$\sum_{p=1,\ldots,P} \sum_{m=2,4,\ldots,M} \sum_{c=1,\ldots,C} t_{dpv}x_{dpmc} \le 1,$$
$$d = 1,\ldots,2A, \quad v = 1,\ldots,V, \quad (7)$$

$$\sum_{p=1,\ldots,P} \sum_{m=2,4,\ldots,M} \sum_{c=1,\ldots,C} t_{dpv}x_{dpmc}$$
$$= \sum_{p=1,\ldots,P} \sum_{m=2,4,\ldots,M} \sum_{c=1,\ldots,C} u_{\tau(d)pv}x_{\tau(d)pmc},$$
$$d = 1,\ldots,A, \quad v = 1,\ldots,V, \quad (8)$$

$$\sum_{p=1,\ldots,P} \sum_{g=1,\ldots,G} z_{dgp} = 1, \quad d = 2A+1,\ldots,D. \qquad (9)$$

The goal of optimization is to minimize the total width of the spectrum, defined as the number of slices occupied whenever in the network, which is reflected in (1). The constraint (2) guarantees that the traffic volume in each optical channel selected for demand d will not exceed the channel capacity. The whole demand must be served, which yields (3). Each slice in each link can be used only for one channel, which is expressed by (4). The constraint (5) defines that slice s is employed in the network when it is used in any of the network links. The constraint (6) assures that the computational power of data centers is not exceeded. For each anycast demand maximally one path to and from each data center may be used, which is exhibited by (7). Each anycast upstream stream must be associated with the corresponding anycast downstream stream, which implies (8). Finally, the constraint (9) ensures that exactly one path is assigned to each unicast demand.

3.3. Average spectrum optimization: Problem formulation.

The average spectrum optimization (AvSO) problem (Markowski, 2015) consists in minimizing the average width of the spectrum used in the network. The width of spectrum used in each link is represented by the largest index of allocated slot. We consider the following problem:

objective

$$\text{minimize } AvS = \frac{1}{E} \sum_{e=1,\ldots,E} q_e \qquad (10)$$

subject to (2)–(4), (6)–(9),

$$sy_{es} \le q_e, \quad e = 1,\ldots,E, \quad s = 1,\ldots,S. \qquad (11)$$

The objective (10) is to minimize the average spectrum usage, defined as the average number of slices occupied in network links. The constraint (11) defines the largest index of allocated slices in link e. The remaining constraints have the same meaning as in the optimization problem with the objective ToS.

Both the formulated optimization problems are NP-hard (Klinkowski and Walkowiak, 2011). The proposed ILP formulation can be used for obtaining optimal solutions with tools for solving linear optimization problems, e.g., CPLex (ILOG, 2016), for small problem instances. For big network topologies and a high number of demands, obtaining the optimal solution is impossible in reasonable time and with reasonable computing resources (memory, processing power). Then, in the remainder of paper, we propose heuristic approaches to the problem considered: heuristic algorithms able to find a feasible, sub-optimal solution

and a meta-heuristic based on the tabu search method, allowing us to obtain results closer to optimality.

4. Heuristic algorithms

In this section, we propose construction-type greedy algorithms to obtain fast suboptimal solutions. The advantage of such algorithms is the short computing time, but the obtained results may be far from optimal.

The idea is to process demands in the selected order and assign a lightpath (or a few lightpaths) to each demand (or stream). In each iteration of the algorithm, one selected demand is processed. One of three ordering methods of selecting demands in consecutive iterations may be chosen:

- Anycast first (AF). The algorithm starts with allocating anycast upstream demands in the decreasing order of the demand volume. After allocating all anycast upstream demands, unicast demands, also in volume decreasing order, are allocated. Finally, the algorithm allocates anycast downstream demands, since their volumes and the required number of slices are smallest (for each anycast downstream demand, an optical channel with the width of two slots is allocated).

- Unicast first (UF). The algorithm starts with allocation of unicast demands, followed by anycast upstream demands, and then anycast downstream demands. Unicast and anycast upstream demands are processed in the decreasing order of the demand volume, as in the AF method.

- Biggest first (BF). Unicast and anycast upstream demands are considered concurrently in the decreasing order of the demand volume. After that, anycast downstream demands are considered.

Let B be the set of demands that must be allocated. The AF heuristic algorithm (Algorithm 1) starts from set B containing all anycast upstream demands. In each iteration of the algorithm, one demand with the highest unsent volume of data is chosen (Step 2). Let Q_d be the set of possible paths for part of demand d. Each anycast demand may be directed to many data centers. We assume that for each anycast demand for each data center only one path and one channel may be chosen in order to send the part of the demand directed to this data center (we call the flow allocating this part of the demand a 'stream'). The number of data centers and the ratio of demand streams are the decision variables.

Each anycast upstream demand is divided into parts, and those are allocated in consecutive iterations. The volumes of parts of the demand are equal (the last part may be smaller) to the amount of data that may be sent during one second through the channel built with two

Algorithm 1. Anycast first algorithm.

Step 1. $B \leftarrow \{d : d = 1, 2, \ldots, A\}$.

Step 2. $d^* = \arg \max(h_d : d \in B), H = h_d^*$.

Step 3. If $(H \geq 4\Delta_s)$ then $\varepsilon = 4\Delta_s$; otherwise, $\varepsilon = H$.

Step 4. For each path $p \in Q_d$ do $\Phi_p = FS(d^*, p)$.

Step 5. $p = \arg \min(\Phi_p)$, if the minimal value of Φ was obtained for more than one path, use the path length, in terms of the number of links, as an additional criterion. Allocate part ε of demand d^* to path p^*.

Step 6. $H = H\varepsilon$. If $H > 0$ then go to Step 3.

Step 7. $B \leftarrow B \setminus \{d^*\}$; if $B \neq \emptyset$ then go to Step 2.

Step 8. $B \leftarrow \{d : d = 2A + 1, 2A + 2, \ldots, D\}$.

Step 9. $d^* = \arg \max(h_d : d \in B)$.

Step 10. Allocate demand d^* to path $p \leftarrow FQ(d^*)$.

Step 11. $B \leftarrow B \setminus \{d^*\}$; if $B \neq \emptyset$ then go to Step 9.

Step 12. $B \leftarrow \{d : d = A + 1, A + 2, \ldots, 2A\}$.

Step 13. For each $d \in B$ and for each stream of $\tau(d^*)$ allocate the stream of demand d^* to path $p \leftarrow FQ(d^*)$.

slices with a minimal (equal to one) modulation level. As assumed in Section 3, the capacity of such a channel is equal to 4Δ, where Δ denotes the frequency slice width. The volume ε of part of demand is calculated in Step 3. There are two allocation options for a successive part of one demand: a new stream may be created or the part considered may be added to one of the already existing streams. A new stream may be created only if there is a data center not used by the given demand. If the part of demand is to be added to an existing stream, the channel for that stream must be exchanged with a bigger one. For such an operation an appropriate number of free slices, adjacent to the existing channel, may be needed. If there is no possibility of creating a new stream, nor adding the part of demand to any of the existing streams, then one of the existing streams must be reallocated to the new channel, allowing accommodation of the given part of demand.

Let function $FS(d, p)$ return the value of the degradation of the objective function in the case of allocating part of demand d on path p, starting from the lowest accessible slice. Moreover, $FS(d, p)$ verifies the available processing power of the data center located at the end of path p, and returns $+\infty$ in the case of insufficient computational resources. All the previously allocated demands and parts of demand d are taken into account when calculating the value of $FS(d, p)$. For those data centers which are not used for demand d, we take into account all paths leading to the data center node. For the already used data centers, we take into account only the path to which the existing stream is allocated. The path with the lowest value of $FS(d, p)$ is then used for allocating part of demand d (Step 5). In case there

are a few paths with the same value of FS, the length of the path, in terms of the number of links, is taken into account as an additional criterion. For each unicast demand, a single channel leading from the source node to the destination node must be allocated (Steps 8–11).

Let the function $FQ(d)$ return the index $p*$ of the path for demand d, with the lowest value of the degradation of the objective function in the case of allocating demand d to path $p*$, starting from the lowest accessible slice. All the previously allocated demands are taken into account when calculating the value of $FQ(d)$. The path given by $FQ(d)$ is then used for allocating demand d. In case there are a few paths with the same value of degradation, the path length, in terms of the number of links, is taken into account as an additional criterion. Anycast downstream demands also may be realized with a few streams. The downstream stream of downstream anycast demands must be allocated for each data center utilized for the associated upstream demand. We assume that each downstream stream requires the minimal possible channel capacity. Then it is built using two slices. For downstream streams, we choose the best path using the criterion function $FS(d, p)$ (Step 13).

In the heuristic algorithm with the UF selecting method, Steps 1–7 and 8–11 are swapped. In the case of the heuristic algorithm BF, the AF algorithm should be modified in the following way:

Step 1. $B \leftarrow \{d : d = 1, 2, \ldots, A, 2A + 1, \ldots, D\}$. Then, in Step 2, demand d^* is selected for allocation. If $d^* \leq A$, then Steps 3–6 are executed, otherwise Steps 9–10 are executed.

5. Tabu search algorithm

Tabu search is one of meta-heuristic methods widely used to solve many kinds of optimization problems (Talbi, 2009; Glover and Kochenberger, 2003). Meta-heuristiscs offer capabilities to find solutions much better than simple heuristics (often close to optimal), and need much fewer resources and less computing time than exact algorithms (e.g., solvers). In this section we propose a tabu search algorithm for the optimization problem considered.

Let Ω be the current solution and Ω^* be the best-known solution. Tabu search (Algorithm 2) starts with an initial solution Ω_0, which must be a feasible solution to the problem considered. It may be obtained from heuristic algorithms (used during experiments) or generated randomly. In the first step of the algorithm, the initial solution becomes the current solution Ω and the best-known solution Ω^*. Then, in each iteration, a neighbor Ω' of the current solution is generated. If the value of the objective function of Ω' is better than the value of the objective function for the current solution, then Ω' becomes the current one. Moreover, if Ω' is better than Ω^*, it also becomes the best-known solution.

The operation of changing the current solution is called a move. In order to avoid multiply testing the same solution, a tabu list is utilized. All tested solutions (or all performed moves) are added to the tabu list. When choosing the neighbor, solutions (moves) from the tabu list are forbidden. The above steps make it possible to find a sub-optimal solution to the problem considered. To escape from local optima, the non-improving solutions are accepted for a specified number of iterations after reaching a sub-optimal solution. This process is called diversification, and it makes it possible to change the search region into a non-explored area. There are a few mechanisms that control the tabu search process: the short-term and long-term memories, diversification attributes, a selection method of neighboring solutions (move), and stopping criteria.

5.1. Solutions. A solution is represented by a sequence of allocations of demands and demand streams, and the paths selected for each demand (or demand stream). Demands (streams) are allocated in the network in sequence order. An optical channel for each demand (or stream) starts from the lowest possible slice. Then, the decision variables are the position of demand (stream) on the allocation sequence, the path for demand (stream), and the volume of the stream (only for anycast upload demands).

5.2. Moves. The aim of the move is to convert the current solution into its neighbor with a better value of the objective function. The neighbor of a given solution Ω is the solution that differs from Ω in the value of one decision variable. According to the problem formulation and solution format, three kinds of moves may be made:

- Changing the position of one demand or stream in the allocation sequence. The move is defined with the following set of attributes: (i) demand number d, (ii) stream (data center) number r—for anycast demands, and (iii) the shift within the allocation sequence (positive or negative).

- Choosing another path for demand or stream. The move is defined with the following set of attributes: (i) demand number d, (ii) stream (data center) number r—for anycast demands, and (iii) current path number p_{cur} and new path number p_{new}.

- Changing the volume of two anycast upstream streams. The 4Δ [Gbps] flow may be reallocated from one of the streams existing for demand d (source stream) to another stream of this demand (destination stream). As a result of this kind of move, a new stream (to a previously unused data center) may be created or an existing stream may be eliminated (with an update of the associated

downstream stream). Besides the impact on network flows, this kind of move reallocates the part of computations (proportional to reallocated traffic) from one (source) data center to another (destination) data center, which is possible only when sufficient processing power is available in the destination data center—constraints put on processing power are verified at this step. A move is defined with the following set of attributes: (i) demand number d, (ii) source stream (data center) number k_{sour} and destination stream number k_{dest}, and (iii) the volume of flow moved from k_{sour} to k_{dest}.

In our further discussion we will denote the move as mv and the neighbor of solution Ω generated as a result of move mv as $(\Omega + mv)$.

For each current solution, we determine the set $\Gamma(\Omega)$ of priority moves. Priority moves are nominated on the basis of the properties of the current solution in the following way. For an optimization problem with the AvS objective function, for each network link e we calculate the number of wasted slices. Slice s is wasted when it is not assigned to any optical channel and there are assigned slices with higher slice index on link e. Next, for each link with a non-zero number of wasted slices we check which demand d and stream number r (when d is anycast demand) occupy the highest slice used on that link. All moves, i.e.,

- lowering the position of demand d (and stream r if applicable) on the allocation sequence,
- changing the path for demand d (and stream r if applicable),
- reallocating part of demand d from stream r to another stream (only for anycast upstream demands),

are added to set $\Gamma(\Omega)$.

For example, consider the following slice occupation on link e. Slices 1–4 are assigned to demand d_1, Slices 7–8 are assigned to demand d_2 and Slices 11–20 are assigned to demand d_3. Then the number of wasted slices for link e is equal to 4 (Slices 5, 6, 9, 10 are not used), and the appropriate moves for demand d_3 (occupying the highest slice used with index 20) will be added to the set of priority moves. For the optimization problem with objective ToS, we use another definition of priority moves. We find the highest allocated slice s^* over all network links. Then, for each link e with occupied slice s^*, we check which demand d and stream number r (when d is anycast demand) occupy slice s^*. Next, all moves defined for the ToS objective are added to set $\Gamma(\Omega)$. Priority moves are considered in the first order while choosing the neighbor Ω'. Moves are chosen randomly from set $\Gamma(\Omega)$. If there are no possible moves in set $\Gamma(\Omega)$ (i.e., all moves from set $\Gamma(\Omega)$ are on the tabu list) the next move is chosen from the moves not belonging to $\Gamma(\Omega)$.

5.3. Short-term tabu list. The idea of short-term memory is to prevent the testing of already tested solutions while searching the neighborhood of the current solution (Talbi, 2009) . All tested candidate moves which did not yield a better solution are stored in a short-term tabu list (STT). The length of the short-term tabu list is unlimited. This list is flushed when a better solution is found and the move is performed.

5.4. Long-term tabu list. The idea of long-term memory is to store information on the visited solutions found during the search (Talbi, 2009). All performed moves are added to the long-term tabu list (LTT). The length (TL) of the long-term tabu list is one of the key parameters of an algorithm and has to be carefully selected in order to get the best performance and best results of an algorithm (Glover and Kochenberger, 2003).

5.5. Diversification parameters. Diversification is the process that makes it possible to change the search area after reaching the local optima. There are two important parameters of diversification. The first one is the number of iterations without a solution improvement, which is denoted as threshold DT (diversification threshold). After DT iterations with solution improvement, we decide that a local optimum has been found. Then the tabu algorithm starts to accept the neighboring solutions worse than the current one, in order to escape from local optima. DT is given as a parameter. Its value may have a strong impact on the tabu algorithm performance and the quality of solutions, so it should be carefully selected in the tuning process. The linear parameter DR (diversification ratio) defines the level of acceptance of worse solutions. Let $F(\Omega)$ be the value of the objective for solution Ω. A solution Ω' will be accepted during diversification if $F(\Omega') < F(\Omega) \times (1 + DR)$.

5.6. Stopping criteria. The following three stopping conditions were used: the time limit, the iteration limit, and founding an optimal solution. The time limit is given in seconds. Each assignment of a candidate move is counted as iteration, including neighborhood candidate move generation and diversification moves. When the number of iterations exceeds the iteration limit, the algorithm terminates. The third stopping condition (reaching the optimal solution) can be applied only when an optimal solution for the problem considered is known in advance (i.e., it has been obtained from the CPLEX solver, which is possible for very small problem instances). All stopping conditions are optional, but at least one of them must be applied.

5.7. Tabu search algorithm. Let I be the iteration counter of the tabu algorithm, and I_{ni} be the counter of

iterations with no solution improvement (non-improved). Θ is the diversification coefficient; the algorithm starts

---

Algorithm 2. Tabu search algorithm.

1: $\Theta := 1, ;\ I := 0, ;\ I_{ni} := 0$
2: $STT := \emptyset, ;\ LTT := \emptyset$
3: $\Omega^* := \Omega := \Omega_0$
4: determine the set of priority moves $\Gamma(\Omega)$
5: **repeat**
6:     select possible non-tabu move mv
7:     **if** $ToS(\Omega + mv) < \Theta \times ToS(\Omega)$ **then**
8:         $\Omega := \Omega + mv$
9:         determine the set of priority moves $\Gamma(\Omega)$
10:         $STT := \emptyset$
11:         $LTT := LTT \cup \{mv\}$
12:         $\Theta := 1$
13:         $I_{ni} := 0$
14:     **else**
15:         $STT := STT \cup \{mv\}$
16:     **end if**
17:     **if** $ToS(\Omega) < ToS(\Omega^*)$ **then**
18:         $\Omega^* := \Omega$
19:     **end if**
20:     $I++$
21:     $I_{ni}++$
22:     **if** $((\Omega = 1)$ **and** $(I_{ni} > DT))$ **then**
23:         $\Theta := \Theta + DR$
24:     **end if**
25: **until** stopping criteria satisfied
26: **return** Ω^*

---

with Θ equal to 1. Both I and I_{ni} start from the value zero at the beginning (Line 1); also, the short term tabu list and the long term tabu list are empty (Line 2). The initial solution Ω_0 is used as the current solution (Line 3). For the current solution we determine the set of priority moves (Line 4). Then, in each iteration, a move is generated. Moves are generated randomly. Before selecting move mv for use in the algorithm, it is checked if mv is not in set STT or LTT. If, as a result of move mv, a solution with the criterion value smaller than the current one has been generated (Line 7), the new solution becomes the current one. Moreover, if the the algorithm is in the diversification phase, the value of Θ is greater than 1 and worse solutions are also acceptable (to escape from local minima). If the move is accepted for execution, the newly generated solution becomes the current one (Line 8), the set of priority moves for the new solution is determined (Line 9), the short term tabu list becomes empty (Line 10), and the move mv is added to the long term tabu list (Line 11).

Finally, the value 1 for the diversification coefficient and the value 0 for the number of iterations with no improvement are restored (Lines 12–13). If the selected move mv is not beneficial (and is not performed, Line 14), it is added to the short-term tabu list (15). If the criterion

value of the current solution is better than the criterion value for the best solution Ω^*, we tag the current solution as best (Lines 17–19). The number of iterations is incremented in Line 20, and the number of solutions without an improvement in Line 21. Line 22 is the diversification check. If the number of iterations without improvement is bigger than diversification threshold DT, the value of coefficient Θ is increased with the value of the diversification ratio (DR). The algorithm terminates when the stopping criteria are satisfied (iteration or time limit, best solution found). Ω^* is the result of the tabu algorithm.

6. Experiments and results

In this section results of the experiments for both the ToSO and AvSO problems are reported. The main goals of the experiments were to evaluate the quality of the solutions obtained with the proposed heuristic algorithms, including tabu search, and to analyze the spectrum usage for different data center allocation scenarios. Experiments were conducted for four network topologies: NFS15, EURO16, EURO28 and UBN24 (cf. Fig. 2).

The first part of the experiments focused on tuning the tabu search algorithms for both the problems. The aim of the tuning was to choose the best values of the length of the long-term tabu list (TL), the diversification threshold (DT) and the diversification ratio (DR). NFS15 and EURO16 topologies with 5 Tb/s overall demands in the network were examined during this part of the experiments. In the second part of the experiments, the quality of heuristic algorithms was studied. The optimal solutions, obtained with the CPLEX solver (ILOG, 2016),
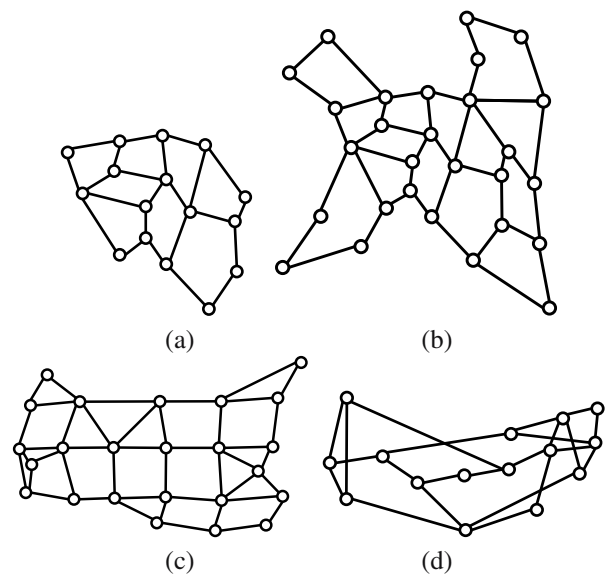

Fig. 2. Network topologies: EURO16 (a), EURO28 (b), UBN24 (c), and NFS15 (d).

constituted a basis for estimation. Due to the complexity of the problems considered, optimal solutions were found for relatively small network topologies and a small volume of demands, two network topologies NFS15 (Fig. 2(d)) and EURO16 (Fig. 2(a)) were examined. The overall demand in the network was set to 2 Tb/s for the ToS optimization problem and 1 Tb/s for the AvS optimization problem. For bigger networks and bigger values of overall demands in the network (20 Tb/s for NFS15, 25 Tb/s for EURO16, 50 Tb/s for EURO28 and 40 Tb/s for UBN24), tabu search results were compared with the heuristic AF, UF and BF algorithms. Finally, we investigated the impact of the choice of the problem parameters on the solution for different data center allocation scenarios.

The sets of demands were generated randomly (source and destination nodes for unicast demands, client nodes for anycast demands, demand volumes). The volume of anycast upstream demands was chosen from the range 40 to 400 Gbps, the volume of anycast downstream (feedback) demands was always set to 10 Gbps, and the volume of unicast demands was selected from the range the 10 to 100 Gbps. Different anycast to unicast volume ratios were used to generate demand sets.

Let AR be the anycast ratio defined as the proportion of the volume of all anycast demands to the volume of all unicast demands. Demand sets with six different values of AR, were examined: 0% (no anycast traffic), 20%, 40%, 60%, 80%, and 100% (no unicast traffic). For each value of AR three demand sets were considered. In a majority of the experiments (with the exception of those when the impact of the available processing power on the solution quality was investigated) the summarized processing power of all data centers was equal to 150% of the power required to complete all computational tasks, and it was uniformly distributed over all data centers. Since the goal of optimization is to minimize the use of the spectrum, we do not put any limits on the total number of slices available per link. The values of S, chosen for the computational experiments for each scenario, were significantly higher than the largest indexes of allocated slots. Experiments were performed on a computer equipped with an Intel Core i5 2.67 GHz processor and 4 GB RAM.

6.1. Tuning of the tabu search algorithm. There are three tabu search algorithm parameters whose values must be properly selected in order to ensure a good performance of the algorithm and a high quality of the results. They are the length TL of the long-term tabu list, the diversification threshold DT and the diversification ratio DR. In the initial phase of tuning, we selected the most promising range of values of parameters TL, DT and DR. Having observed the best results for $DR = 0.1$, we decided to use this value for all further experiments. For the ToSO

Table 1. Examined TS parameters for the ToSO problem.

Parameter	Examined values	Selected value
TL	5, 8, 11, 14, 17, 20, 23, 26	20
DT	70, 100, 130, 160, 190, 220, 250, 280, 310, 340, 370, 400, 430	340
DR	0.1, 0.2, 0.3	0.1

Table 2. Examined TS parameters for the AvSO problem.

Parameter	Examined values	Selected value
TL	3, 5, 8, 11, 14, 17, 20, 23	8
DT	160, 180, 200, 220, 240, 260, 280, 300	260
DR	0.1, 0.2, 0.3	0.1

problem we observed the best results for TL in the range from 5 to 26, and for DT in the range from 70 to 430, for the AvSO problem the best results were observed for TL in the range 3 to 23, and for DT in the range from 160 to 300. For the main tuning phase we generated demand sets with 5 Tb/s overall demands, one for each $AR = 0\%, 20\%, 40\%, 60\%, 80\%$, and 100%. For each demand set, we examined scenarios with three sets of data center allocation (with 1, 2, and 3 data centers) and three sets of candidate paths (3, 10, and 30 candidate paths for each pair of nodes). Finally, for each of the examined networks and for both optimization problems considered, tuning was performed for 6×3×3=54 different scenarios. The values of the examined parameters for the ToSO are shown in Table 1, while the values for AvSO are listed in Table 2. Experiments were conducted for each combination of TL and DT listed in the tables; 8×13=104 parameter sets for the ToSO problem and 64 parameter sets for the AvSO problem. Since the proposed tabu algorithm is not deterministic, the experiments for each parameter set were performed 10 times and the averaged results were taken into account. Finally, for each examined network topology 56160 experiments for the ToSO problem and 34560 experiments for the AvSO problem were performed. The selected best parameter sets are presented in Tables 1 and 2, respectively.

6.2. Quality of approximate solutions for small networks and small overall demand. The main goal of this part of the experiments was to examine the quality of approximate solutions based on optimal solutions produced by the CPLEX solver. The optimal solutions were obtained for NFS15 and EURO16, the overall demand was set to 2 Tb/s for the ToSO problem and 1 Tb/s for the AvSO problem. The number of candidate paths was set to 2. For both NFS15 and EURO16, the

Table 3. Gap between optimal and approximate solutions for the AvSO problem.

	CPLEX	TS	AF	BF	UF
Mean gap to optimal	0.00%	0.17%	17.8%	17.7%	12.7%
Minimum gap to optimal	0.00%	0.00%	0.00%	0.00%	0.00%
Maximum gap to optimal	0.00%	17.39%	87.14%	87.14%	87.14%
Avarage execution time [s]	468.3	26.1	0.1	0.1	0.1

Table 4. Gap between optimal and approximate solutions for the ToSO problem.

	CPLEX	TS	AF	BF	UF
Mean gap to optimal	0.00%	9.39%	19.05%	18.89%	26.82%
Minimum gap to optimal	0.00%	0.00%	0.00%	0.00%	0.00%
Maximum gap to optimal	0.00%	75.00%	115.38%	115.38%	137.50%
Avarage execution time [s]	27.5	10	0.1	0.1	0.1

experiments were conducted for six values of AR, three sets of demands for each value of AR, three different numbers of data centers, and four sets of nodes of data center allocation for each number of data centers (overall, 216 scenarios for each network).

The average, minimal and maximal gap between optimal solutions and those yielded from the tabu search algorithm and heuristic algorithms, as well as the average computation times for the AvSO problem, are presented in Table 3. For the CPLEX solver the time limit of two hours was set. In most experiments the solver was able to find an optimal solution in less than 15 minutes, but for some scenarios (8%) no optimal solution was found in the fixed time limit. Experiments for the tabu search (TS) algorithm were repeated five times for each scenario. The time limit equal to 300 s and reaching an optimal solution were used as the stopping criteria. In most cases (97%) the tabu search algorithm was able to find an optimal solution, and the average gap between optimal and TS solutions was 0.17%. The average, minimal and maximal gaps and execution times for optimal solutions and those yielded from the tabu search algorithm and heuristic algorithms for the ToSO problem are presented in Table 4. For the CPLEX solver the time limit of two hours was set. For all scenarios optimal solutions were found. The maximal time needed to find an optimal solution was 874 seconds (around 15 minutes); in most experiments the solver was able to find an optimal solution in less than 60 seconds. Experiments for the tabu search (TS) algorithm were repeated five times for each scenario. The time limit equal to 10 seconds was used as a stopping criterion. In 48.94% of the experiments the tabu search algorithm was able to find an optimal solution, and the average gap between optimal and TS solutions was 9.39%.

The problem considered has an off-line nature, which means that all demands are known in advance and last for a relatively long time. Then it is usually acceptable to spend tens of minutes or even hours in order to find the best possible solution. In such cases the tabu

Table 5. Average gap to best solution for the ToSO problem [%].

Network	Volume	AF	BF	UF	TS
NFS15	20 Tb/s	6.59	6.56	42.72	2.06
EURO16	25 Tb/s	7.20	7.20	30.22	2.65
EURO28	50 Tb/s	6.82	6.82	40.50	1.72
UBN24	40 Tb/s	5.92	5.92	52.06	1.51

Table 6. Average gap to best solution for the AvSO problem [%].

Network	Volume	AF	BF	UF	TS
NFS15	20 Tb/s	23.64	23.51	18.55	4.16
EURO16	25 Tb/s	29.64	29.63	21.84	6.14
EURO28	50 Tb/s	19.20	19.20	31.96	2.81
UBN24	40 Tb/s	21.51	25.83	35.99	4.32

search algorithm offers the best agility since it makes it possible to regulate optimization time (using the time limit stopping criterion) and utilize the whole available time for improving the obtained solution as much as possible. We may also imagine large-scale computing systems where changes are required more frequently, and routes must be assigned within seconds or milliseconds—then fast heuristic approaches would be more applicable, at the expense of higher spectrum usage.

6.3. Comparison of heuristics for all networks and large overall demand. For more complex scenarios (bigger networks, bigger demand volumes, bigger number of candidate paths) the CPLEX solver was not able to perform optimization. Then the experiments were conducted with the proposed AF, BF, UF, and tabu search (TS) algorithms. In this section we present the comparison of solutions obtained with the heuristic algorithms for network topologies and the overall demand volume, listed in Table 5 for the ToSO problem and in Table 6 for the AvSO problem. We investigated 732 scenarios for NFS15 and EURO15 networks:

- three demand sets for each $AR = 0\%, 20\%, 40\%, 60\%, 80\%,$ and $100\%,$

- for non-zero AR: $R = 1, 2, 3$ data centers, four sets of data center location for each number of data centers,

- $k = 2, 5, 10, 30$ candidate paths for each pair of network nodes.

For EURO28 and UBN24 networks, 244 scenarios were investigated:

- one demand set for each $AR = 0\%, 20\%, 40\%, 60\%, 80\%,$ and $100\%,$

- for non-zero AR : $R = 2, 3, 4$ data centers, four sets of data center location for each number of data centers,

- $k = 3, 5, 10, 30$ candidate paths for each pair of network nodes.

The TS algorithm was run 10 times for each scenario, the time limit was set to 60 seconds for NFS15 and EURO16, and 12 minutes for EURO28 and UBN24. For each experiment, the distance from the best result was calculated (the best result for each scenario was yielded from the tabu search algorithm). Average distances from the best solution for all experiments for the ToSO problem are presented in Table 5. As we may conclude from the results, the proposed tabu search algorithm overcomes other heuristics. Deteriorated divergence of TS results was observed for a larger number of candidate paths, especially for bigger network topologies. From the other heuristics, AF and BF algorithms produced similar results, while those obtained from the UF algorithm were significantly worse. Average distances from the best solution for all experiments for the AvSO problem are presented in Table 6. The TS algorithm significantly overcomes other heuristics. AF and BF algorithms yield quite a similar solution. In contrast to the ToSO problem, for smaller network topologies the UF algorithm yielded better solutions than AF and BF algorithms.

6.4. Analysis of the influence of problem parameters on the quality of solutions.
For all network topologies the influence of the number of candidate paths (k) on the solutions was examined. We relied on experiment scenarios, values of AR, the number of data centers and the number of candidate paths described in the previous section. We analyzed results for both the ToSO and AvSO problems. All results presented in this section were obtained with the tabu search (TS) algorithm, with parameters chosen in the tuning process and with 60 seconds (for NFS15 and EURO16) or 12 minutes (for EURO28 and UBN24) as time limits applied as the stopping criterion.
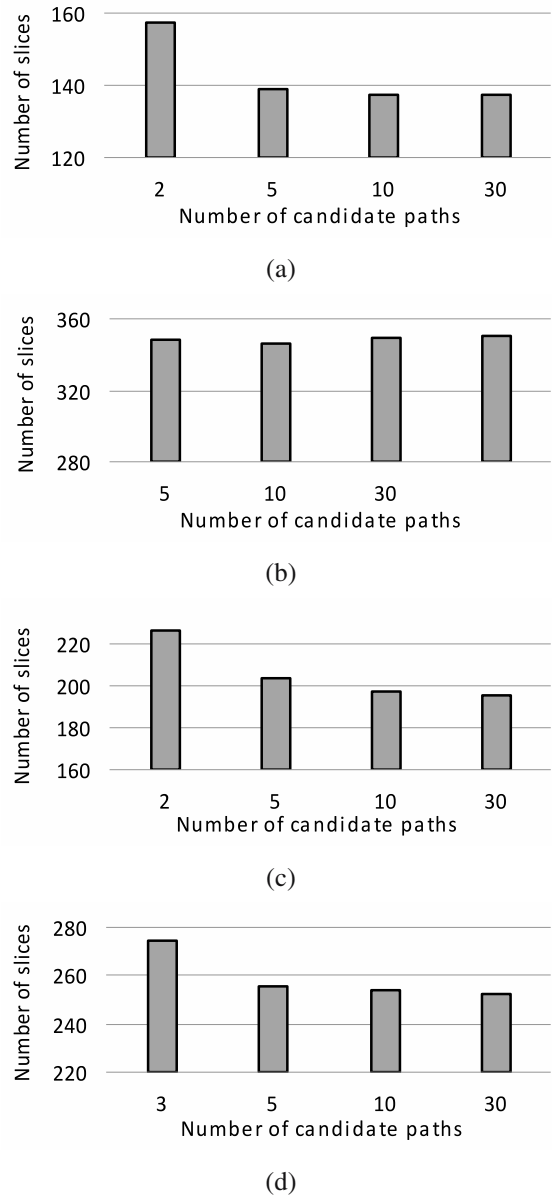

Fig. 3. Influence of the number of candidate paths on ToS: EURO16 (a), EURO28 (b), NFS15 (c) UBN24 (d).

Figure 3 depicts the influence of the number of candidate paths on the criterion function ToS. For NFS15, EURO16 and UBN24, we observed that a larger number of candidate paths makes it possible to find a better solution to the discussed problem. The highest improvement for NFS15 and EURO16 networks was observed where there are from two to five candidate paths, and for the network UBN24 when there are from three to five candidate paths. Further increasing the number of candidate paths brings a slight improvement in ToS. We may conclude that for small networks and the optimization time restricted to 60 seconds, the optimal
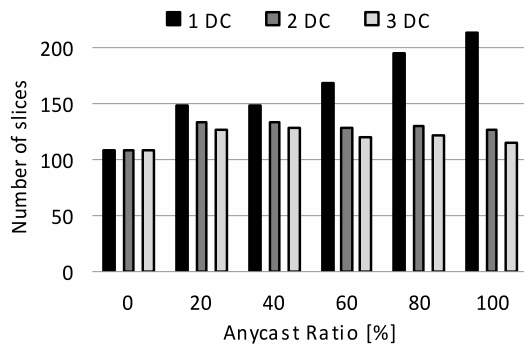
Fig. 4. Influence of the anycast ratio and the number of data centers on ToS for EURO16; total demand volume: 25 Tb/s.
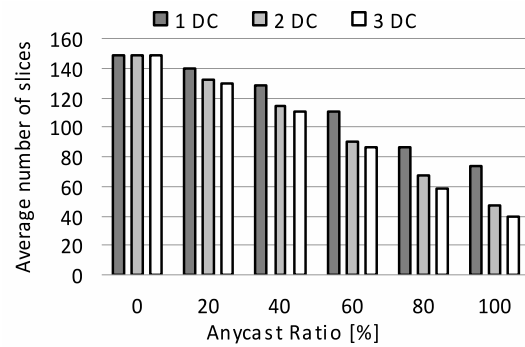

Fig. 6. Influence of the anycast ratio and the number of data centers on AvS for EURO16; total demand volume: 25 Tb/s.
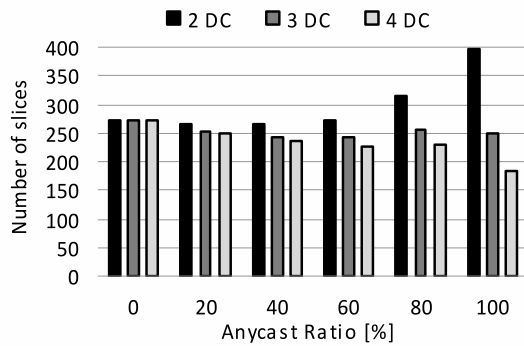

Fig. 5. Influence of the anycast ratio and the number of data centers on ToS for UBN24; total demand volume: 40 Tb/s.
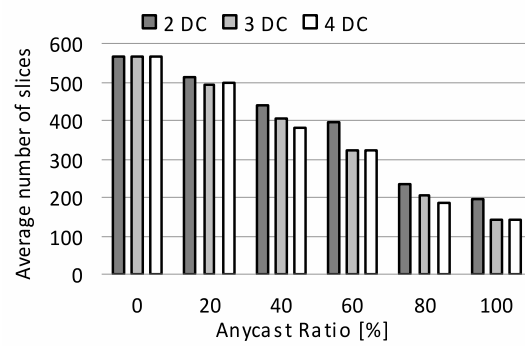

Fig. 7. Influence of the anycast ratio and number of data centers on AvS for EURO28; total demand volume: 50 Tb/s.

number of candidate paths is equal to 10. For EURO28 we observed a minimal impact of the number of candidate paths on the solution. A slight improvement in ToS was observed for $k = 5$, but a further increase degraded the solution. A likely reason for this is that the tabu search algorithm was not able to search the entire solution space during the time limit. For bigger networks, using sets of candidate paths containing more than five paths for each node pair is not beneficial if the optimization time is an important criterion. A higher number of paths increases the time needed to solve the problem considered and the tabu algorithm is not able to search the whole solution area, which may have a negative impact on the quality of solutions. We also observed a high ratio of moves without any improvement. It may be concluded that the TS algorithm does not scale well for the increasing number of candidate paths in big networks.

The impact of the value of the AR coefficient and the number of data centers on solutions for ToSO and AvSO was analyzed. For scenarios with one data center we observed a strong influence of the anycast ratio on the ToS criterion. Results for EURO16 (Fig. 4) show that the width of the spectrum required for $AR = 100\%$ doubled in comparison with $AR = 0\%$. For two and three

data centers the required spectrum width is significantly lower than for single data center scenarios. Moreover, for larger values of AR we observed that increasing the number of data centers from two to three results in a small improvement in the objective value. Taking into account the cost of establishment and maintenance of additional data centers, we assume that two data centers for small network topologies might be a consensual choice. We also observed that for small networks (EURO16, NFS15) the values of objectives were minimal for scenarios with no anycast traffic (user to data center), despite the same overall demand volume in all experiments. For bigger networks (results for UBN24 are presented in Fig. 5), we observed a high impact of the number of data centers on the objective, especially for $AR \geq 60\%$. For $AR = 100\%$, the total spectrum required for the distributed computing system composed of four data centers is half the spectrum width required for a two-data-center system. It is also worth noticing that for EURO28 and scenarios with only unicast traffic ($AR = 0$), the system requires more spectrum resources than for a higher ratio of anycast traffic (if more than 2 data centers were used).

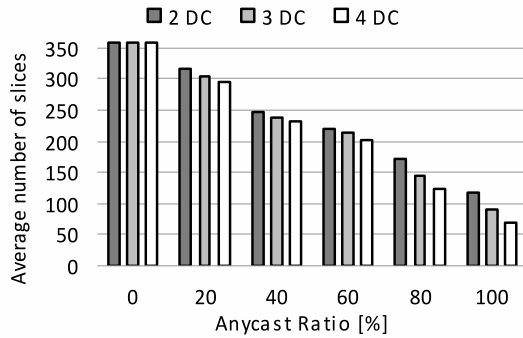Experiments for the AvSO problem (Figs. 6–8) show that the average width of the spectrum decreases when

Fig. 8. Influence of the anycast ratio and the number of data centers on AvS for UBN24; total demand volume: 40 Tb/s.

Fig. 9. Influence of the available processing power on ToS for EURO28.

increasing the value of the anycast ratio. Also, the number of data centers has a significant impact on AvS. For EURO16 the highest difference in the value of AvS was observed between scenarios with one and two data centers, for EURO28 and UBN24 between two and three data centers. A slighter improvement in AvS may be obtained when increasing the number of data centers from two to three for small networks and from three to four (for bigger ones). The observed dependencies may be helpful in making decisions about the optimal number of data centers that should be placed in the EON-based large-scale computing system in order to achieve the most spectrum efficient and cost efficient solution.

Let the processing power ratio (PPR) be the ratio of the summated available processing power of all data centers to the power required to complete all computational tasks. The impact of the available processing power on ToS for EUR28 is presented in Fig. 9. Experiments were carried out for $AR = 70\%$, the overall demand volume equal to 50 Tb/s, and the processing power uniformly distributed over all data centers. It was observed that the required width of the spectrum rapidly increases when the amount of spare processing power falls below 30%. On the other hand, expanding the processing power over 170% of PPR does not significantly improve the solution. We also noticed during the experiments that the TS algorithm scales quite well when increasing the processing power of data centers.

6.5. Case study. Data center location in large-scale computing systems may impact both spectrum usage and ownership cost. One of the important kinds of operation efforts is the cost of energy used in data centers. In this section we present an analysis of different data center location strategies. Experiments were performed with the TS algorithm for the EURO28 topology, based on the European wide area network. Scenarios with two and four data centers were investigated; the details of data center location for each scenario are listed in Table 7.
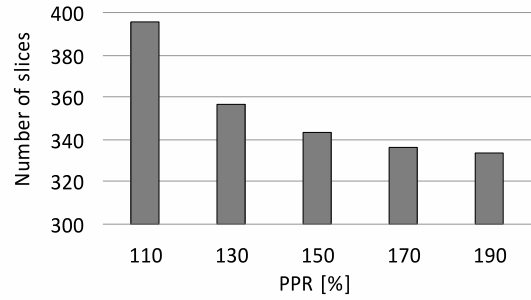
For the first two of them (s1_2DC, s2_2DC for two data centers and s1_4DC, s2_4DC for four data centers) the potentially most beneficial locations were chosen. Data centers were located in the nodes in the center of wide area networks, with a high number of neighboring nodes. For the remaining scenarios, the nodes located in the cities with the lowest energy prices were chosen. The aim of the study was to discover whether cost efficient allocation of data centers over the European network involves the highest utilization of spectrum resources.

In our study, $AR = 60\%$ and the overall demand volume equal to 50 Tb/s were set. The AvS optimization problem was investigated. The results of experiments are shown in Fig. 10; spectrum utilization was juxtaposed with the mean price of energy used in data centers. Let $r = 1, \ldots, R$ be data centers in a large-scale computing system. Let w_r be the number of instructions executed in data center r (in GIPS) and v_r be the price of energy in data center r (in EUR/MWh). The mean price of energy (MPE) for the obtained solutions was calculated as

$$\mathrm{MPE} = \frac{\sum_{r=1}^{R} w_r v_r}{\sum_{r=1}^{R} w_r}. \tag{12}$$

As expected, the allocation of data centers in central nodes makes it possible to save spectrum resources. For 2 data center scenarios (Fig. 10(a)), location strategies s1 and s2 make it possible to allocate all demands using, on average, 363 and 341 slices, respectively, while with cost efficient locations spectrum utilization increases up to 453 (scenario s3) and 452 (scenario s4). The benefit is a considerably lower cost of energy, the mean price is equal to 64 EUR/MWh for scenarios s3 and s4, and 89 EUR/MWh for scenario s2 and up to 113 EUR/MWh for s1. Similar dependencies were observed while studying results of experiments with four data centers (Fig. 10(b)). For scenarios s1 and s2 lower spectrum utilization was observed, while scenarios s3 and s4 ensure a remarkable reduction in energy cost.

Table 7. Case study scenarios.

Scenario	Number of data centers	Data center locations
s1_2DC	2	London, Rome
s2_2DC	2	Berlin, Brussels
s3_2DC	2	Paris, Belgrade
s4_2DC	2	Strassbourg, Belgrade
s1_4DC	4	London, Rome, Munich, Hamburg
s2_4DC	4	London, Berlin, Milan, Budapest
s3_4DC	4	Amsterdam, Belgrade, Bordeaux, Lyon
s4_4DC	4	Amsterdam, Belgrade, Strassbourg, Paris
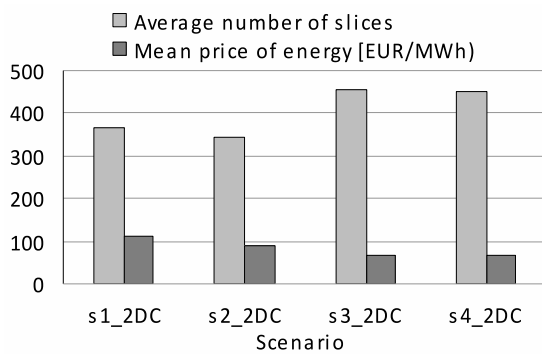
(a)
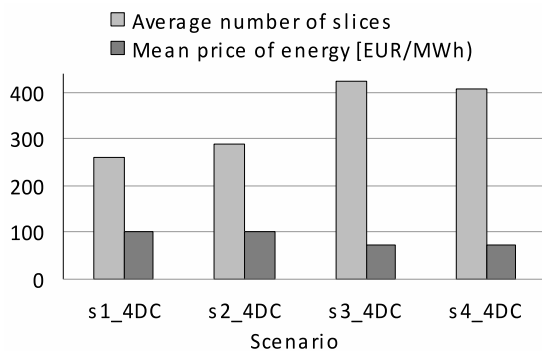
(b)

Fig. 10. Case study: two data centers (a), four data centers (b).

7. Conclusion

In the paper the problem of optimization of unicast and anycast traffic in an elastic optical network-based large-scale computing systems has been considered. The original spectrum optimization problem that consists in computational tasks allocation among data centers, route selection and spectrum allocation in an EON has been formulated. Two indicators of spectrum usage were analyzed as objectives in the optimization process. The optimization problem is new, as it assumes limits on the computational power of data centers, and the possibility of dividing tasks into subtasks for execution on many data centers. The problem considered is NP-hard, being more general than the RSA problem. We have proposed an ILP formulation which allows us to find an optimal solution with CPLEX solver for less complex scenarios, three heuristic greedy algorithms and a meta-heuristic algorithm based on the tabu search method. The main novelty of the proposed tabu search algorithm is the method of assigning priority moves for each tabu solution.

Intensive computer experiments have been conducted with the proposed algorithms for both objectives. We have examined 4 network topologies and over 6,000 scenarios, and have conducted over 200,000 experiments with the tabu search algorithm. Results of the experiments show that the proposed tabu search algorithm significantly outperforms simple heuristics and in a significant number of scenarios (for small networks) is able to produce an optimal solution to the problem considered, while for the remaining examined scenarios the average gaps from the optimum do not exceed 10%. It has also been observed that the values of problem parameters: the number of data centers, the available processing power and the anycast ratio have a considerable impact on the spectrum usage in the EON-based large-scale computing system. The dependencies analyzed in the paper may be helpful to make decisions about the optimal number and power of data centers as well as decisions concerning optimal balancing between spectrum usage and cost of computation.

References

Aibin, M. and Walkowiak, K. (2015). Regenerator placement algorithms for cloud-ready elastic optical networks, *Proceedings of the 17th International Conference on Transparent Optical Networks, ICTON 2015, Budapest, Hungary*, pp. 1–4.

Chatterjee, B.C., Sarma, N. and Oki, E. (2015). Routing and spectrum allocation in elastic optical networks: A tutorial, *IEEE Communication Surveys and Tutorials* **17**(3): 1776–1800.

Christodoulopoulos, K., Tomkos, I. and Varvarigos, E. A. (2011). Elastic bandwidth allocation in flexible OFDM-based optical networks, *Journal of Lightwave Technology* **29**(9): 1354–1366.

Cierniak, R. (2014). An analytical iterative statistical algorithm for image reconstruction from projections, *International Journal of Applied Mathematics and Computer Science* **24**(1): 7–17, DOI: 10.2478/amcs-2014-0001.

Cisco (2014). Cisco white paper, Cisco visual networking index: Forecast and methodology, 2013–2018, *Technical report*, Cisco Systems Inc., San Jose, MA.

Fallahpour, A., Beyranvand, H., Nezamalhosseini, S.A. and Salehi, J.A. (2014). Energy efficient routing and spectrum assignment with regenerator placement in elastic optical networks, *Journal of Lightwave Technology* **32**(10): 2019–2027.

Fang, W., Lu, M., Liu, X., Gong, L. and Zhu, Z. (2015). Joint defragmentation of optical spectrum and it resources in elastic optical datacenter interconnections, *IEEE/OSA Journal of Optical Communications and Networking* **7**(4): 314–324.

Glover, F. and Kochenberger, G. (2003). *Handbook of Meta-heuristics*, Kluwer Academic Publishers, Boston, MA.

Goścień, R., Walkowiak, K. and Klinkowski, M. (2014). Distance-adaptive transmission in cloud-ready elastic optical networks, *Journal of Optical Communications and Networking* **6**(10): 816–828.

Goścień, R., Walkowiak, K., Klinkowski, M. and Rak, J. (2015). Protection in elastic optical networks, *IEEE Network* **29**(6): 88–96.

ILOG (2016). IBM ILOG CPLEX documentation, www-01.ibm.com.

ITU-T (2012). ITU-T recommendation G.694.1 (Ed. 2.0), Spectral grids for WDM applications: DWDM frequency grid.

Jinno, M., Takara, H., Kozicki, B., Tsukishima, Y., Sone, Y. and Matsuoka, S. (2009). Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies, *IEEE Communications Magazine* **47**(11): 66–73.

Klinkowski, M. and Walkowiak, K. (2011). Routing and spectrum assignment in spectrum sliced elastic optical path network, *IEEE Communications Letters* **15**(8): 884–886.

Layec, P., Morea, A., Vacondio, F., Rival, O. and Antona, J.C. (2013). Elastic optical networks: The global evolution to software configurable optical networks, *Bell Labs Technical Journal* **18**(3): 133–151.

Liu, X., Gong, L. and Zhu, Z. (2013). Design integrated RSA for multicast in elastic optical networks with a layered approach, *Proceedings of the IEEE Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA*, pp. 2346–2351.

Markowski, M. (2015). Tabu-search algorithm for optimization of elastic optical network based distributed computing systems, *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2015, Wrocław, Poland*, pp. 361–369.

Talbi, E. (2009). *Metaheuristics: From Design to Implementation*, John Wiley and Sons, Hoboken, NJ.

Talebi, S., Alam, F., Katib, I., Khamis, M., Salama, R. and Rouskas, G.N. (2014). Spectrum management techniques for elastic optical networks: A survey, *Optical Switching and Networking* **13**: 34–48.

Tornatore, M., Rottondi, C., Goścień, R., Walkowiak, K., Rizzelli, G. and Morea, A. (2015). On the complexity of routing and spectrum assignment in flexible-grid ring networks, *Journal of Optical Communications and Networking* **7**(2): A256–A267.

Velasco, L., Klinkowski, M., Ruiz, M. and Comellas, J. (2012). Modeling the routing and spectrum allocation problem for flexgrid optical networks, *Photonic Network Communications* **24**(3): 177–186.

Walkowiak, K., Kasprzak, A. and Klinkowski, M. (2014a). Dynamic routing of anycast and unicast traffic in elastic optical networks, *Proceedings of the IEEE International Conference on Communications, ICC 2014, Sydney, Australia*, pp. 3313–3318.

Walkowiak, K. and Klinkowski, M. (2013). Joint anycast and unicast routing for elastic optical networks: Modeling and optimization, *Proceedings of the IEEE International Conference on Communications, ICC 2013, Budapest, Hungary*, pp. 3909–3914.

Walkowiak, K., Klinkowski, M., Rabiega, B. and Goścień, R. (2014b). Routing and spectrum allocation algorithms for elastic optical networks with dedicated path protection, *Optical Switching and Networking* **13**: 63–75.

Wang, M., Liu, J. and Zhou, W. (2016). Design and implementation of a high-performance stream-oriented big data processing system, *Proceedings of the 8th Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016, Hangzhou, China*, pp. 363–368.

Wang, Y., Cao, X. and Pan, Y. (2011). A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks, *Proceedings of IEEE INFOCOM 2011, Shanghai, China*, pp. 1503–1511.

Warren, M.S., Brumby, S.P., Skillman, S.W., Kelton, T., Wohlberg B., Mathis M., Chartrand R., Keisler R., and Johnson M. (2015). Seeing the earth in the cloud: Processing one petabyte of satellite imagery in one day, *Proceedings of the Applied Imagery Pattern Recognition Workshop, AIPR 2015, Washington, DC, USA*, pp. 1–12.

Wubin, L., Svrd, P., Tordsson, J. and Elmroth, E. (2013). Cost-optimal cloud service placement under dynamic pricing schemes, *Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC 2013, Dresden, Germany*, pp. 187–194.

Yoo, S.J.B. and Wen, K. (2012). Intra and inter datacenter networking: The role of optical packet switching and flexible bandwidth optical networking, *Proceedings of the International Conference on Optical Network Design and Modeling, ONDM 2012, Colchester, UK*, pp. 1–6.

Yoo, S.J.B., Yin, Y. and Proietti, R. (2013). Elastic optical networking and low-latency high-radix optical switches for future cloud computing, *Proceedings of the International Conference on Computing, Networking and Communications, ICNC 2013, San Diego, CA, USA*, pp. 1097–1101.

Zhang, L. and Zhu, Z. (2014). Dynamic anycast in inter-datacenter networks over elastic optical infrastructure, *Proceedings of the International Conference on Computing, Networking and Communications, ICNC 2014, Honolulu, HI, USA*, pp. 491–495.

Zhu, Z., Lu, W., Zhang, L. and Ansari, N. (2013). Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing, *Journal of Lightwave Technology* **31**(1): 15–22.

Marcin Markowski was born in Wrocław, Poland, in 1977. In 1996 he started MSc studies in computer science at the Wrocław University of Technology. He received his MSc and PhD degrees in computer science from that university in 2001 and 2005, respectively. Currently he holds a scientific position there. His research interests include planning and optimization of computer networks, virtualization and distributed systems. He has been involved in a number of research projects. He is a program committee member of several international conferences as well as a reviewer for international journals.