A MILP-Based Heuristic for Energy-Aware Traffic Engineering with Shortest Path Routing

Edoardo Amaldi, Antonio Capone, Luca G. Gianoli, and Luca Mascetti

DEI, Politecnico di Milano, Piazza L. da Vinci 32, Italy
`last_name@elet.polimi.it`

Abstract. Internet energy consumption is rapidly becoming an issue due to the exponential traffic growth and the rapid expansion of communication infrastructures worldwide. We address the problem of energy-aware intra-domain traffic engineering in networks operated with a shortest path routing protocol. We consider the problem of switching off (putting in sleeping mode) network elements (links and routers) and of adjusting the link weights so as to minimize the energy consumption as well as maximizing a measure of effectiveness of the routing weight configuration. We propose a three-phase MILP-based heuristic for tackling this multi-objective problem with priority (first minimize the energy consumption and then the overall cost of link utilization), which exploits the IGP-WO heuristic proposed for optimizing the link weights so as to minimize the total cost of link utilization. For comparison purposes, we also developed a greedy randomized search procedure with path-relinking. The computational results for four real network topologies and different types of traffic matrices show that it is possible to switch off a substantial number of core nodes during low and moderate traffic periods, while guaranteeing the same point-to-point service quality and moderately increasing the network total cost of link utilization.

1 Introduction

Data reported in [9] show that in 2007 Internet had been responsible for 5.5% of the total energy consumption in the world and that the annual increment rate can be estimated around 20-25%. For these reasons, the issues of energy saving in IP networks and of power awareness in network design have recently become of great interest in the scientific community and have attracted the interest of device manufacturers and Internet Service Providers (ISP).

Energy management in the Internet exploits the fact that networks are designed and dimensioned to serve the estimated peak traffic demand. Usually, during network operation, traffic load varies remarkably over time and even during peak hours it is usually well below network capacity. Unfortunately, current network device architectures and transmission technologies make their power consumption almost independent of the traffic load. As a result, networks consume energy as if they were always fully loaded.

We consider the most widely used Internal Gateway Protocol (IGP) in IP networks, namely the Open Shortest Path First (OSPF) protocol. Traffic demands are routed from origin to destination along the shortest paths computed with respect to the weights assigned to the links. If the equal cost multi-path (ECMP) rule is considered, the packets

J. Pahl, T. Reiners, and S. Voß (Eds.): INOC 2011, LNCS 6701, pp. 464–477, 2011.

are evenly split at nodes where more outgoing links belong to shortest paths to the destination. Link weights are managed by network operators, who may modify them in order to optimize routing and reduce network congestion.

Let the directed graph $G = (V,A)$ represent the network topology, where V is the set of nodes and A the set of links. We distinguish two types of nodes: *edge* nodes and *core* nodes. Edge nodes can be both source and destination of traffic demands, while core nodes play only the role of transit routers. Let D denote the traffic matrix, where d_{ij} is the traffic demand for each pair of edge nodes i and j, and $d_{ij} = 0$ for all other pairs of nodes.

We consider the following extension of the IGP weight optimization problem with ECMP rule for intra-domain Traffic Engineering, that we refer to as *Energy-aware Traffic Engineering* (E-TE).

E-TE: Given a directed graph $G = (V,A)$, representing the topology of an IP network composed by routers and links with capacities on the links, and a traffic matrix D, decide which network elements (routers and links) to switch off and determine the link weights so as to minimize the total network energy consumption (primary objective) and a measure of effectiveness of the routing weight configuration (secondary objective), while guaranteeing that all the traffic demands are routed and the maximum utilization constraint is satisfied for each link.

According to the distinction between primary and secondary objectives, we first look for a sub-network with minimum total energy consumption and then minimize the total cost of link utilization proposed in [6] on the sub-network corresponding to the active elements. The unnecessary routers and links can then be excluded from the shortest path trees, by assigning a very large value to the corresponding link weights. Note that by switching off a node or a link we do not necessarily mean to turn it completely off but to put it in sleeping mode.

After a summary of related work in Section 2, a mixed integer programming formulation is presented in Section 3. Since it is very challenging even for small size instances, in Section 4 we propose a three-phase heuristic for tackling the E-TE problem combining greedy procedures with the iterative solution of a relaxed MILP-formulation for a sub-network with increasing traffic matrices. Building on previous work on intra-domain traffic engineering, we use the IGP-WO algorithm proposed by Fortz and Thorup [6] that, given a network topology and a traffic matrix, aims at a set of link weights that minimizes the total cost of link utilization. For comparison purposes we have also implemented a greedy randomized procedure (GRASP) with path-relinking. In Section 5 we report an discuss the computational results obtained on four real network topologies with different types of traffic matrices. Finally, Section 6 contains some concluding remarks.

2 Related Work

Since the seminal work by Gupta and Singh [8], the research community has started developing technologies for manufacturing energy efficient network devices, methodologies for power aware network design, and energy management strategies for reducing energy wastes of networks in operation.

To the best of our knowledge, there are only a few recent works on energy-aware traffic engineering.

The approach proposed in [16] aims at switching off the line-cards (network links) guaranteeing QoS constraints (maximum utilization and maximum path length constraints) in a scenario where an hybrid MPLS/OSPF scheme is adopted. The approach is based on a MIP formulation where the traffic demands are routed through a set of k-shortest path previously calculated.

In [4] the authors describe some heuristics that, given a traffic matrix and a fully powered network, are able to switch off nodes and links while respecting traffic constraints. In [15] some Energy-Aware Traffic Engineering (EATe) techniques are presented for optimizing links and routers power consumption, by considering their rate-dependant energy profiles. Assuming that the energy consumption of the network elements depends on the different rates, EATe algorithms try to switch off the underutilized elements or to reduce their rate, by re-routing part of the traffic in other network portions without increasing the rate of any element. Unlike in our work, the approaches in [4] and [15] follow a flow-based strategy that is suitable for the Internet geographic backbone where label switching protocols are adopted and not for routing domains based on shortest paths.

The energy management algorithm for IP networks called Energy Aware Routing (EAR) algorithm and presented in [5] is able to switch off network elements exploiting a modified version of the OSPF protocol. EAR algorithm selects a subset of routers, Importers Router (IR), that do not calculate their own shortest path tree (SPT) but use that of some neighboring routers, Exporters Routers (ER). In general a small number of active SPTs reduces also the number of links used that can be switched off. There are several important differences between our work and [5]. As in the literature on OSPF traffic engineering [13], we keep the OSPF protocol unchanged and focus on optimizing the link weights, while in the EAR algorithm the weights are assumed to be given and the protocol needs to be modified to implement ERs and IRs. Moreover, we explicitly consider link capacity limitations and minimize the network congestion level in order to guarantee service quality, while in [5] neither traffic load nor network capacity are clearly considered.

The reader is referred to [3] and [10] for summaries of the work done over the past decade on link weights optimization for intra-domain traffic engineering. Different objective functions have been considered (e.g., link utilization cost function minimization [6], residual capacity maximization and load balancing maximization) and different heuristic methods have then been developed (e.g., local search, genetic algorithm, Lagrangian approach). The code of the well-known IGP-WO algorithm [6] is available from the TOTEM toolbox [2]. To the best of our knowledge, the issue of energy-aware link weights optimization has not yet been addressed.

3 Mixed Integer Programming Formulation

Let p_{ij} and p_k be the power consumption of link (i,j) and node k respectively. Let c_{ij} be the capacity of the link (i,j). If the binary decision variables x_{ij} and y_k represent the power status (on/off) of links and routers respectively, a Mixed Integer Linear Programming (MILP) formulation of the part of the E-TE problem involving the the energy consumption minimization is given by:

$$\min \sum_{(i.j)\in A} p_{ij}x_{ij} + \sum_{k\in V} p_k y_k \tag{1}$$

s. t.

$$x_{ij} \le y_i \qquad (i,j)\in A \tag{2}$$

$$x_{ij} \le y_j \quad (i,j)\in A \tag{3}$$

$$\sum_{i\in V} f^t_{it} = \sum_{s\in V} d_{st} \quad t\in N \tag{4}$$

$$\sum_{j\in V} f^t_{vj} - \sum_{i\in V} f^t_{iv} = d_{vt} \quad v,t\in V, t\neq v \tag{5}$$

$$\sum_{t\in V} f^t_{ij} \le x_{ij}\alpha c_{ij} \quad (i,j)\in A \tag{6}$$

$$0 \le z^t_i - f^t_{ij} \le (1-u^t_{ij})\sum_{v\in V} d_{vt} \quad t\in V,(i,j)\in A \tag{7}$$

$$f^t_{ij} \le u^t_{ij}\sum_{v\in V} d_{vt} \quad t\in V,(i,j)\in A \tag{8}$$

$$0 \le r^t_j + \omega_{ij} - r^t_i \le (1-u^t_{ij})M \quad t\in V,(i,j)\in A \tag{9}$$

$$1-u^t_{ij} \le r^t_j + \omega_{ij} - r^t_i \quad t\in V,(i,j)\in A \tag{10}$$

$$u^t_{ij} \le x_{ij} \quad (i,j)\in A, t\in V \tag{11}$$

$$\omega_{ij} \ge (1-x_{ij})\omega_{max} \tag{12}$$

$$1 \le \omega_{ij} \le \omega_{max} \quad (i,j)\in A \tag{13}$$

$$\omega_{ij} \in Z \quad (i,j)\in A \tag{14}$$

$$u^t_{ij} \in \{0,1\} \quad t\in V,(i,j)\in A \tag{15}$$

$$x_{ij}, y_k \in \{0,1\} \quad (i,j)\in A, k\in V \tag{16}$$

$$f^t_{ij} \ge 0 \quad (i,j)\ \in A, t\in V \tag{17}$$

$$r^t_i, z^t_i \ge 0 \quad i,t\in V, \tag{18}$$

where M is a large enough constant. The objective function (1) aims at minimizing the total energy consumption of the network. Constraints (2)-(3) ensure that if a node is switched off all incident links are turned off. Obviously a node can be switched off only if there are no traffic demands having it as source or destination (edge or core node). Constraints (4)-(5) are the classical flow conservation constraints, where the (real) positive variable f_{ij}^t indicates the amount of flow routed through the link $(i, j) \in A$ destined to node $t \in N$. Constraints (6) are the maximum utilization constraints imposing that the total flow through each link does not exceed the link maximum utilization and forcing the flow to 0 if the link (i, j) is powered off; the parameter α is comprised between 0 and 1. The binary variables $u_{ij}^t = 1$ appearing in Constraints (7)-(9) describe the routing configuration: $u_{ij}^t = 1$ if and only if the link (i, j) belongs to one of the shortest paths from node i to node t. Constraints (7) make sure that if $u_{ij}^t = 1$ then the flow f_{ij}^t destined to node t is equal to the (real) variable z_i^t, which is the common value of the flow assigned to all links outgoing from i and belonging to the shortest paths from i to t. Constraints (8) force $f_{ij}^t = 0$ for all links (i, j) that do not belong to a shortest path to node t. Finally, the shortest path routing Constraints (9)-(15) assure that the routing vector u defines shortest paths consistent with the link weight vector ω and forbid switched off links to belong to a shortest path; moreover the switched off links weights are put equal to the maximum value w_{max}. For each pair of nodes j and t, the (real) variable r_j^t corresponds to the length of the shortest path from node j to node t.

Unfortunately, the above MILP-formulation, which is an extension of the one given in [10] for intra-domain traffic engineering, turns out to be very challenging even for small size networks. For instance, we did not manage to find a feasible integer solution for a small network with 10 nodes and 42 links in 10 hours of computing time.

4 Heuristic Algorithms

4.1 Greedy Algorithm with Dual Weights Initialization (GA-DW)

A first simple approach to tackle the E-TE problem is to adopt a greedy strategy. Given a network topology G and a traffic matrix D, an initial set of links weights with low total cost of link utilization is obtained by applying the IGP-WO algorithm (see below for the initialization). Then we sort the network elements (nodes and links) according to some intuitive criteria, consider them in that order and try to switch off as many of them as possible.

We use three criteria for sorting nodes. In Least-Link (LL), Least-Flow (LF), and Sum-of-Weights (SW), nodes are sorted in non-decreasing order according to, respectively, the degree (number of incident links), the total amount of traffic flowing through them, and the sum of the weights of all the incident (active) links. We consider two criteria for sorting links. In Least-Flow (LF) and Traffic-Engineering (TE), links are sorted in non-decreasing order according to, respectively, the total amount of traffic flowing through them and their weight. Two of these criteria, Least-Link (LL) and Least-Flow (LF), were already used in a different context in [4]. The six combined node-link sorting policies are given in Table 1. The nodes are always considered before the links because of their higher energy consumption.

Table 1. Combinations of sorting criteria, the rows correspond to the link criteria and the columns to the router criteria

	LF	**LL**	**SW**
LF	LF-LF	LL-LF	SW-LF
TE	LF-TE	LL-TE	SW-TE

At each step of the greedy procedure, we verify whether the next available active network element according to the sorting order can be turned off. The considered element is actually turned off if the OSPF routing determined on the reduced network by the link weights of the active links, is able to support the traffic matrix (all traffic demands) without exceeding the link maximum utilization limit α comprised between 0 and 1. A run terminates when all the network elements have been tested.

The initial set of link weights is determined by applying 150 iterations of IGP-WO with a maximum weight value of 100. Since the java implementation of IGP-WO in the TOTEM toolbox [2] is computationally heavy, we speed up the procedure by following the suggestion in [3]. The idea is to warm-start IGP-WO with the set of link weights obtained with the procedure described in [14], namely by taking as initial link weights the values of the dual variables of the following linear programming multicommodity flow relaxation:

$$\min \quad \Phi = \sum_{(i,j)\in A} \phi\left(c_{ij}, l_{ij}\right) \tag{19}$$

s. t.

$$\sum_{j\in V} f^t_{vj} - \sum_{j\in V} f^t_{jv} = \begin{cases} -\sum_{s\in V} d_{st} & if \quad v = t \\ d_{vt} & if \quad v \neq t \end{cases} \qquad (v,t) \in N \tag{20}$$

$$l_{ij} = \sum_{t\in V} f^t_{ij} \qquad (i,j) \in A \tag{21}$$

$$\phi\left(c_{ij}, l_{ij}\right) \geq \alpha_z l_{ij} - \beta_z c_{ij} \qquad (i,j) \in A, z \in Z \tag{22}$$

$$f^t_{ij} \geq 0 \qquad (i,j) \in A, t \in V. \tag{23}$$

The objective function (19) is, as in the IGP-WO algorithm [6], the sum of piecewise linear convex functions $\phi\left(c_{ij}, l_{ij}\right)$ based on the links utilization. Constraints (20) are the classic flow conservation constraints, while Constraints (21) force the total flow on each arc to be equal to the sum of all the flows routed through the arc itself. The dual variables used as link weights are those corresponding to Constraints (21). Constraints (22) define the link utilization cost function. Finally Constraints (23) define the positive flow variables f^t_{ij}, that are equal to the amount of traffic routed through the arc (i, j) and destined to node t.

In the *Greedy Algorithm with Dual Weights Initialization* (GA-DW) the greedy procedure is run six times (once for each combined node-link sorting policy) and the best solution found is returned. Finally, a set of link weights for the resulting sub-network is obtained by executing 150 iterations of IGP-WO so as to reduce the total cost of link utilization.

Note that the set of values of the link weights is determined only twice: at the beginning for the initial network topology and at the end of the algorithm for the resulting sub-network.

4.2 Two-Stage Algorithm with Dual Weights Initialization (TA-DW)

Since the problem of finding a minimum energy sub-network of G with an optimized set of link weights is very challenging, we split it into two stages. The TA-DW procedure includes a switching-off stage and a feasible routing stage.

The *Switching-off Stage*, which receives as input the complete network topology G and the given traffic matrix denoted by D, aims at selecting the set of network elements that could be switched off. This is achieved by solving within a 3% gap the following Mixed Integer Linear Program (MILP) that is a subset of the E-TE formulation (1)-(18):

$$\min \sum_{(i.j)\in A} p_{ij}x_{ij} + \sum_{k\in V} p_{k},y_k \tag{24}$$

s. t.

$$x_{ij} \le y_i \quad (i,j) \in A \tag{25}$$

$$x_{ij} \le y_j \quad (i,j) \in A \tag{26}$$

$$\sum_{i\in V} f_{it}^{t} = \sum_{s\in V} d_{st} \quad t \in N \tag{27}$$

$$\sum_{j\in V} f_{vj}^{t} - \sum_{i\in V} f_{iv}^{t} = d_{vt} \quad v,t \in V, t \neq v \tag{28}$$

$$\sum_{t\in V} f_{ij}^{t} \le x_{ij}\alpha c_{ij} \quad (i,j) \in A \tag{29}$$

$$x_{ij}, y_k \in \{0,1\} \quad (i,j) \in A, k \in V \tag{30}$$

$$f_{ij}^{t} \ge 0 \quad (i,j) \in A, t \in V. \tag{31}$$

The objective function (24) aims at minimizing the network energy consumption. Constraints (25)-(31) are identical to Constraints (2)-(6) and (16)-(17) of the E-TE formulation. Since the node power consumption p_i is in general much larger (about ten times) than the link power consumption p_{ij}, the formulation will give the priority to switching off the nodes. Note that the traffic demands routing is considered fully splittable, see Constraints (31). This formulation falls within the well-known class of capacitated multi-commodity minimum cost flow problems (CMCF) [7].

The *Feasible Routing Stage*, which receives as input the sub-network $G(D)$ determined at the *Switching-Off Stage*, aims at finding a set of link weights that allows to route through the sub-network all the traffic demands according to shortest paths, without exceeding the link maximum utilization α $(0 < \alpha \leq 1)$. The link weights configuration of the second stage is determined by applying 150 iterations of the IGP-WO algorithm with a maximum weights value of 100. As for GA-DW, IGP-WO is warm-started using as input the dual weights computed by solving (19)-(23) for the reduced network $G(D)$ determined at the *Switching-Off Stage*.

Unfortunately, given the sub-network topology $G(D)$ obtained at the first stage and its respective traffic matrix D, there is no guarantee that there exists a set of link weights allowing feasible routing of all traffic demands. This may occur because the first stage considers a fully splittable routing that can be hardly reproduced by the OSPF protocol. In case no feasible OPSF weights set exists (or is found), we slightly increase the original traffic matrix D by multiplying it with a fixed parameter γ, and we repeat the first stage with the increased traffic matrix $D(\gamma)$. γ is equal to 1 at the first iteration, and is increased by 0.1 every time IGP-WO fails to find a feasible set of link weights. If the maximum utilization level is greater than α but smaller than $\alpha + 0.01$, the γ parameter is increased by 0.05 rather than 0.1. This operation clearly leads to a sub-network $G(D(\gamma))$ with more active elements as input of the second stage (the second stage is always run with the original traffic matrix D). The *Feasible Routing Stage* ends when a feasible set of link weights is found.

To check whether some other elements of the resulting sub-network can still be switched off, we apply a last run of the GA-DW algorithm.

4.3 MILP-Based Algorithm

To achieve high quality solutions in a reasonable computing time, we combine the two above algorithms. The idea is to achieve a trade-off between the very low computational load of GA-DW and the robustness of TA-DW.

The resulting *MILP-based Algorithm* (MILP-BA) is composed of the following two stages:

- A complete run of GA-DW with a maximum utilization level of $\alpha - 0.1$.
- A complete run of TA-DW, where the nodes switched-off at the first stage, are forced to remain off. Moreover, the traffic matrix increase criteria are slightly changed, i.e., the γ scaling parameter that is set to 1 at the first iteration, is increased by 0.05 if the maximum utilization level obtained by IGP-WO is comprised between α and $\alpha + 0.1$, or increased by 0.1 if the maximum utilization exceeds $\alpha + 0.1$.

On the one hand, GA-DW allows us to reduce the computational load of TA-DW by substantially reducing the number of variables in the MILP formulation. On the other hand, by decreasing the maximum utilization α by 0.1 in the first run of GA-DW, we tend to avoid that GA-DW forces TA-DW to switch off the wrong network elements.

Table 2. Rocketfuel network topologies

Network	Type	Nodes	Links	$\mathbf{Edge}_{node}$	$\mathbf{Core}_{node}$	$\%\mathbf{Core}_{node}$
Ebone	Backbone	87	322	31	56	64.4
Exodus	Backbone	79	294	38	41	51.9
Sprint	Access	52	168	52	0	0
AT&T	Access	115	296	115	0	0

5 Computational Experiments

5.1 Network Topologies and Traffic Matrices

We have carried out computational tests on four real network topologies provided by the Rocketfuel project [1]. Since our algorithms aim at switching-off both nodes and links, the main focus is on backbone networks that contain edge nodes as well as core nodes and whose core nodes may be switched-off. However, we have also considered access networks that only contain edge nodes, which cannot be switched-off. The characteristics of the four network topologies are summarized in Table 2. The two access networks, Sprint and AT&T, have been used in [16] for testing other energy-aware traffic engineering approaches. Unfortunately, although the network topologies are known, no accurate information is available concerning link capacities and network equipments. For the backbone networks, we assume that all the network links have the same capacity, and we equip each node with routers M10i (power consumption p_i of $86.4W$), and each link with a Gigabit Ethernet line card (power consumption p_{ij} of $7.3W$). Since a router M10i can support at most eight Gigabit Ethernet line cards, the number of routers in each node directly depends on the degree g_i of the node ($\lceil \frac{8}{g_i} \rceil$ routers in each node). For the access networks, we use the capacity values and the equipment configuration kindly provided to us by the authors of [16]. In the case of backbone networks, also the information on the subdivision between edge and core nodes is missing. Since edges nodes can be both source and destination of traffic demands and core nodes play only a role of transit routers, core nodes are the only one that can be powered off. We have randomly selected a set of edge routers for each one of the three network topologies. To avoid the trivial and unrealistic cases where core leaf nodes can be easily powered off, all the leaf nodes are considered as edge nodes. At least one edge node has also been selected for each city.

As to the traffic matrices, for access networks we have used the same traffic matrices as in [16]; the matrices have been obtained by multiplying with different scaling factors the basic matrices computed with the gravity model. For the backbone networks, we have generated the traffic matrices in two different ways:

1. **Constant and Poisson:** generated using the Totem toolbox [2], the maximum load matrices with constant and Poisson traffic distribution that can be supported by the (complete) networks with OSPF *hop-count* routing.

2. **LP-based multicommodity flow matrices:** generated by scaling with a parameter $\beta \in (0,1)$ a maximum supported traffic matrix obtained with a linear programming (LP) formulation. By maximum supported traffic matrix, we mean that all the traffic demands can only be satisfied by switching on all the network links/routers and performing fully splittable routing.

For the sake of simplicity, in our tests we have considered the maximum utilization parameter $\alpha = 1$.

5.2 Results

The computational experiments have been carried out on an Intel Pentium Duo 3.0GHz with 3.5GB of RAM. The results are reported in Tables 3 and 5 for backbone networks and in Table 4 for access networks. The first column indicates the instance considered (network and matrix). Ex, Eb and Spr are the abbreviations for, respectively, Exodus, Ebone and Sprint, while Letters C and P, and numbers (30-40-50 in backbone cases, 7-12-14-21-24-36 in access cases) that follow the networks acronyms correspond to the traffic matrix considered (C for constant matrices, P for Poisson matrices, 30-40-50 for the maximum supported LP matrices scaled by 0.3, 0.4 and 0.5, 7-12-14-21-24-36 for the basic gravity matrices scaled by 7, 12, 14, 21, 24 and 36). Note that by multiplying the Sprint basic matrix by 36 and the AT&T basic matrix by 21, the maximum utilization level obtained performing the OSPF routing is above 90%. The columns $C-E$, L, $E_c^{tot}\ (W)$, $Cong_{min}$ report respectively, the number of core and edge network nodes, the number of network links, the energy consumption of the complete network and the optimized congestion obtained with the complete network. As measure of the congestion level we use the value of the cost function defined by IGP-WO, namely the total cost of link utilization. The remaining columns correspond to the solutions returned by the algorithms; $E_c\ (W)$ is the energy consumption, gap is the ratio $(E_c - E_c^b)/E_c^b$, where E_c^b is the bound on the energy consumption. Note that this energy consumption bound value is calculated by considering the energy consumption of the optimum solution of the MILP formulation for the switching-off stage. $Cong$ indicates the solutions congestion level, $Cong_{\%}$ reports the ratio $Cong/Cong_{min}$. N_{off} and L_{off} show respectively the number of nodes and links switched-off, while t and t_{nor} are respectively the total computing time and the computing time normalized w.r.t. the computing time of GA-DW.

For comparison purposes, we have also adapted the general Greedy Randomized Adaptive Search Procedure (see e.g. [11]) to the E-TE problem. At each iteration of the greedy algorithm, the network element to be switched off is randomly selected among the first $k\%$ elements of the ordered list derived from the sorting criterion. This randomized greedy procedure is run with the same sorting criterion for a predefined number of iterations and the best solution obtained is returned as an approximate solution. We have also endowed our GRASP procedure with a path-relinking feature that allows to intensify the search between elite solutions [12]. Unfortunately, in this case path-relinking only slightly improved the solution quality.

In Table 3 we compare the results obtained running GA-DW and the corresponding version GA-RW with random weights initialization on the backbone networks. The results clearly confirm the impact of this type of link weight initialization. GA-DW results are better in nine cases out of ten.

Table 3. Computational results. Comparison between GA-RW (Greedy Algorithm with Random Weights) and GA-DW on the backbone networks.

				GA-RW				**GA-DW**				**Bound**
Inst	$C-E$	L	$Cong_{min}$	$E_c\,(W)$	N_{off}	L_{off}	$Cong$	$E_c\,(W)$	N_{off}	L_{off}	$Cong$	$E_c^b\,(W)$
Ex30	41-38	294	155052	5146.1	31	169	381870	5239.8	30	168	354534	4546.2
Ex40	41-38	294	253240	5883.5	25	139	499842	5753.3	26	145	566605	5131.5
Ex50	41-38	294	382600	6620.9	19	109	761000	6418.9	21	113	616497	5536.7
ExC	41-38	294	147639	5130.3	30	183	388132	5058.5	31	181	386235	4537.7
ExP	41-38	294	164764	5440.6	27	176	382164	5346.8	28	177	354508	4653.3
Eb30	56-31	322	111265	5677.9	34	207	306940	5677.9	34	207	300163	5540.4
Eb40	56-31	322	169811	6364.2	28	184	353066	6248.6	29	188	369041	5872.6
Eb50	56-31	322	242613	7324.3	19	159	537422	7136.9	21	161	462120	6327.7
EbC	56-31	322	179798	6616.1	25	185	310032	6313.1	28	191	379940	5865.3
EbP	56-31	322	202439	6537.0	26	184	377949	6320.4	28	190	423698	5865.3

Table 4. Computational results obtained with MILP-BA for the Sprint and AT&T access networks

					MILP-BA				
Inst	$C-E$	L	$E_c^{tot}\,(W)$	$Cong_{min}$	$E_c\,(W)$	L_{off}	$t\,(s)$	$Cong$	$Cong_{\%}$
Spr12	0-52	168	24972	28785	11950 (47.9%)	85 (50.6%)	578	160774	558%
Spr24	0-52	168	24972	59651	13339 (53.4%)	76 (45.2%)	584	476098	798%
Spr36	0-52	168	24972	96214	13795 (55.2%)	73 (43.5%)	1241	412256	428%
AT&T7	0-115	296	43344	38990	30504 (70.4%	82 (27.7%)	1816	215903	553%
AT&T14	0-115	296	43344	77980	31026 (71.6%)	79 (26.7%)	1854	323802	415%
AT&T21	0-115	296	43344	117347	32388 (74.7%)	70 (23.6%)	1990	616849	525%

In Table 5 we report the results obtained running GA-DW, G-GA-DW (the adaptation of GRASP to E-TE), TA-DW and MILP-BA on the backbone networks.

The solutions obtained with G-GA-DW are of slightly better quality than those provided by GA-DW but computing times are much higher. In one case the normalized computing time t_{nor} reaches even the value of 20. This is due to the multi-start strategy where the greedy is repeated 100 times for each one of the sorting policies.

TA-DW is very heavy computationally (its normalized computing time t_{nor} reaches the value of 15 in the worst cases) but the quality of the solutions is much better than those found by G-GA-DW. The gap is small, in average smaller than 5%. However, a computing time of 2 up to 8 hours is needed to solve the Ebone instances with up to 87 nodes (56 core nodes and 322 links).

MILP-BA turns out to achieve a remarkable trade-off. Computing times are much smaller than those of TA-DW (maximum value of t_{nor} of 6, and usually smaller than 2), while the solution quality remains the same (except for Ex40). Computing times are generally less than half-hour, and only about half of the time is used to solve the MILP formulations.

Table 5. Computational results. Comparison between GA-DW and G-GA-DW, TA-DW and MILP-BA on the backbone networks. The bound on the energy consumption obtained by solving the MILP formulation is also reported.

					GA-DW								**G-GA-DW**								**Bound**
Inst	$C-E$	L	$E_c^{tot}\,(W)$	$Cong_{min}$	$E_c\,(W)$	gap	N_{off}	L_{off}	$t\,(s)$	t_{nor}	$Cong$	$Cong_{\%}$	$E_c\,(W)$	gap	N_{off}	L_{off}	$t\,(s)$	t_{nor}	$Cong$	$Cong_{\%}$	$E_c^b\,(W)$
Ex30	41-38	294	9058.2	155052	5239.8	15.26%	30	168	1159	1	354534	229%	5124.2	12.71%	31	172	16224	14.00	409715	264%	4546.2
Ex40	41-38	294	9058.2	253240	5753.3	12.12%	26	145	1146	1	566605	224%	5738.7	11.83%	26	147	21152	18.46	604029	239%	5131.5
Ex50	41-38	294	9058.2	382600	6418.9	15.93%	21	113	1252	1	616497	161%	6519.9	17.76%	20	111	25323	20.23	621392	162%	5536.7
ExC	41-38	294	9058.2	147639	5058.5	11.48%	31	181	767	1	386235	262%	4928.3	8.61%	32	187	14632	19.08	331051	224%	4537.7
ExP	41-38	294	9058.2	164764	5346.9	14.91%	28	177	957	1	354508	215%	5152.2	10.72%	30	180	15329	16.02	305352	185%	4653.3
Eb30	56-31	322	10126.6	111265	5677.9	2.48%	34	207	2217	1	300163	270%	5670.6	2.35%	34	208	10121	4.57	334146	300%	5540.4
Eb40	56-31	322	10126.6	169811	6248.6	6.40%	29	188	1897	1	369041	217%	6162.2	4.93%	30	188	13053	6.88	328896	194%	5872.6
Eb50	56-31	322	10126.6	242613	7136.9	12.79%	21	161	1818	1	462120	190%	7107.7	12.33%	21	165	18218	10.02	440984	182%	6327.7
EbC	56-31	322	10126.6	179798	6313.1	7.63%	28	191	1624	1	379940	211%	6298.5	7.39%	28	193	13075	8.05	353153	196%	5865.3
EbP	56-31	322	10126.6	202439	6320.4	7.76%	28	190	1434	1	423698	209%	6226.7	6.16%	29	191	12163	8.48	407699	201%	5865.3

					TA-DW								**MILP-BA**								**Bound**
Inst	$C-E$	L	$E_c^{tot}\,(W)$	$Cong_{min}$	$E_c\,(W)$	gap	N_{off}	L_{off}	$t\,(s)$	t_{nor}	$Cong$	$Cong_{\%}$	$E_c\,(W)$	gap	N_{off}	L_{off}	$t\,(s)$	t_{nor}	$Cong$	$Cong_{\%}$	$E_c^b\,(W)$
Ex30	41-38	294	9058.2	155052	4929.5	8.43%	33	175	4732	4.08	492486	318%	4922.2	8.27%	33	176	2189	1.89	465289	300%	4546.2
Ex40	41-38	294	9058.2	253240	5342.0	4.10%	30	154	10509	9.17	666063	263%	5399.2	5.22%	29	158	2024	1.77	892690	352%	5131.5
Ex50	41-38	294	9058.2	382600	6216.9	12.29%	23	117	10020	8.00	663916	174%	6195.0	11.89%	23	120	4983	3.98	652835	171%	5536.7
ExC	41-38	294	9058.2	147639	4682.5	3.19%	34	197	1939	2.53	465046	315%	4704.4	3.67%	34	194	1863	2.43	441493	299%	4537.7
ExP	41-38	294	9058.2	164764	4820.0	3.58%	33	190	4369	4.57	400072	243%	4805.4	3.27%	33	192	1798	1.88	416809	253%	4653.3
Eb30	56-31	322	10126.6	111265	5670.6	2.35%	34	208	9062	4.09	312596	281%	5569.6	0.53%	35	210	2674	1.21	336005	302%	5540.4
Eb40	56-31	322	10126.6	169811	6111.1	4.06%	30	195	29808	15.71	456886	269%	6096.5	3.81%	30	197	2606	1.37	474118	279%	5872.6
Eb50	56-31	322	10126.6	242613	6689.1	5.71%	25	175	26303	14.47	490739	202%	6667.2	5.37%	25	178	3663	2.01	516547	213%	6327.7
EbC	56-31	322	10126.6	179798	6002.8	2.34%	30	198	9701	5.97	432694	241%	5931.0	1.12%	31	198	1868	1.15	503908	280%	5865.3
EbP	56-31	322	10126.6	202439	6096.5	3.94%	30	197	7657	5.34	490676	242%	6010.1	2.47%	31	197	2648	1.85	423935	209%	5865.3

Concerning the congestion level, it is worth pointing out that in general the total cost of link utilization increases reasonably.

Finally, Table 4 contains the results obtained with MILP-BA for the two access networks. Also in this case a large number of the links can be switched-off (for Spr12 up to 50% of the links). The lower percentage of energy saving achievable for AT&T networks (about half compared with that for Sprint networks) is due to the lower link redundancy and the higher number of leafs that characterize the AT&T topology. The largest computing times are of the order of 30 minutes. Note that, although the percentage increase in congestion is much larger than for backbone networks (up to 8 times), its absolute value is still reasonable.

6 Concluding Remarks

We have investigated the relevant and challenging problem of energy-aware IP traffic engineering with shortest path routing. We have proposed an efficient three-phase MILP-based heuristic which aims at minimizing the energy consumption as well as the total cost of link utilization. The computational results for two real network topologies and different types of traffic matrices show that it allows to switch off a substantial number of core nodes during low and moderate traffic periods, while guaranteeing the same point-to-point service quality and reasonably increasing the network total cost of link utilization.

We leave as future work the extension to account for single link failure and uncertainty in the traffic matrices.

References

1. Rocketfuel Project, `www.cs.washington.edu/research/networking/rocketfuel`
2. Totem Projec, `http://totem.run.montefiore.ulg.ac.be`
3. Altın, A., Fortz, B., Thorup, M., Ümit, H.: Intra-domain traffic engineering with shortest path routing protocols. 4OR: A Quarterly Journal of Operations Research 7(4), 301–335 (2009)
4. Chiaraviglio, L., Mellia, M., Neri, F.: Reducing power consumption in backbone networks. In: IEEE International Conference on Communications, ICC 2009, pp. 1–6. IEEE, Los Alamitos (2009)
5. Cianfrani, A., Eramo, V., Listanti, M., Marazza, M., Vittorini, E.: An energy saving routing algorithm for a green OSPF protocol. In: INFOCOM IEEE Conference on Computer Communications Workshops, pp. 1–5. IEEE, Los Alamitos (2010)
6. Fortz, B., Thorup, M.: Internet traffic engineering by optimizing OSPF weights. In: Proceedings of IEEE Nineteenth Annual Joint Conference on the IEEE Computer and Communications Societies, INFOCOM 2000, vol. 2, pp. 519–528. IEEE, Los Alamitos (2000)
7. Ghamlouche, I., Crainic, T., Gendreau, M.: Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. Operations Research 51(4), 655–667 (2003)
8. Gupta, M., Singh, S.: Greening of the Internet. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 19–26. ACM, New York (2003)
9. Koomey, J.: Estimating total power consumption by servers in the US and the world. Final Report (2007)

10. Pióro, M., Medhi, D.: Routing, flow, and capacity design in communication and computer networks. Morgan Kaufmann Publishers, San Francisco (2004)
11. Resende, M., Ribeiro, C.: Greedy randomized adaptive search procedures. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 57, pp. 219–249. Springer (2003)
12. Resende, M., Ribeiro, C.: GRASP with path-relinking: Recent advances and applications. In: Ibaraki, T., Nonobe, K., Yagiura, M. (eds.) Metaheuristics: Progress as Real Problem Solvers, pp. 29–63. Springer (2005)
13. Sridharan, A., Guerin, R., Diot, C.: Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. IEEE/ACM Transactions on Networking 13(2), 234–247 (2005)
14. Ümit, H., Fortz, B.: Fast heuristic techniques for intra-domain routing metric optimization. In: Proc. of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, vol. 6 (2007)
15. Vasić, N., Kostić, D.: Energy-aware traffic engineering. In: Proc. of the 1st International Conference on Energy-Efficient Computing and Networking, pp. 169–178. ACM, New York (2010)
16. Zhang, M., Yi, C., Liu, B., Zhang, B.: GreenTE: Power-Aware Traffic Engineering. In: IEEE International Conference on Network Protocols (2010)