Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ARIE M.C.A. KOSTER
ADRIAN ZYMOLKA

Linear Programming Lower Bounds for Minimum Converter Wavelength Assignment in Optical Networks

ZIB-Report 04–41 (December 2004)

Linear Programming Lower Bounds for Minimum Converter Wavelength Assignment in Optical Networks

Arie M. C. A. Koster[1,2] Adrian Zymolka[1]

Abstract

In this paper, we study the conflict-free assignment of wavelengths to lightpaths in an optical network with the opportunity to place wavelength converters. To benchmark heuristics for the problem, we develop integer programming formulations and study their properties. Moreover, we study the computational performance of the column generation algorithm for solving the linear relaxation of the most promising formulation. In many cases, a non-zero lower bound on the number of required converters is generated this way. For several instances, we in fact prove optimality since the lower bound equals the best known solution value.

1 Introduction

The design of an optical network involves three closely related tasks: dimensioning, routing, and wavelength assignment. Given is a physical topology of the network as well as, for every pair of nodes, a demand for a number of lightpaths to be established between these nodes. Transmission capacity at a link is provided by the installation of fibers and Wavelength Division Multiplexing (WDM) systems, whereas switching capacity in the nodes is provided by optical cross connects (OXCs). The network has to be dimensioned and the lightpaths have to be routed in such a way that the capacity consumption by the lightpaths does not exceed the transmission and switching capacities of the links and nodes. The routing has to satisfy various constraints, e.g., survivability requirements. In addition, each lightpath has to be assigned a wavelength on every link it passes, where the number of times a wavelength can be assigned is limited by the WDM systems on the link. Two consecutive links of a lightpath can be assigned a different wavelength only by the placement of a so-called wavelength converter in the intermediate node. We assume that each wavelength converter can translate a single optical channel from any wavelength to any other. The cost of an optical network design consists of the cost of the installed equipment. The optical network design problem is to find a design that minimizes the total cost.

[1]Zuse Institute Berlin (ZIB), Takustraße 7, D–14195 Berlin, Germany. Email: {koster,zymolka}@zib.de

[2]Partially supported by the DFG research group "Algorithms, Structure, Randomness" (Grant number GR 883/9-3, GR 883/9-4)

Keywords: Optical Networks, Wavelength Assignment, Integer Programming

Mathematics Subject Classification (2000): 90C10, 90B18, 90C90

In the literature, the focus of optical network design has been on routing and wavelength assignment, whereas the dimensioning was considered less frequently. For incorporation of all three issues, we have proposed in [13] a decomposition of the problem into a dimensioning and routing subproblem and a wavelength assignment subproblem. In the first subproblem, a cost minimal configuration of fibers, WDM systems, and OXCs has to be found together with an appropriate routing of all lightpaths. In the second subproblem, we have to find a conflict-free wavelength assignment to the routed lightpaths, which can always be carried out due to the unlimited availability of wavelength converters. Computational experiments favor such a decomposition, since the obtained optical network designs typically require only a small number (often zero) of wavelength converters, cf. [6, 13]. The fraction of total network cost spent for conversion remains therefore very small. The decomposition is logical also from a mathematical point of view, as it separates two hard mathematical problems, an integer multicommodity flow problem (routing) and a generalized coloring problem (wavelength assignment).

In this paper, we study the subproblem of assigning wavelengths to the routed lightpaths such as to minimize the number of wavelength converters. In [6], it has been shown that the problem is $\mathcal{NP}$-hard, and heuristics have been discussed. To benchmark the heuristics, it is necessary to compute the optimum or, second best, a good lower bound on the number of converters needed. For this, we present two integer programming formulations and compare the quality of the lower bounds provided by the linear relaxations to each other and to the integer optimum. To compute the linear relaxation optimum of the favorite formulation, a column generation algorithm has been implemented. For several realistic instances, it turns out that this lower bound equals the best known heuristically computed value, by this proving optimality. In a computational study, we further compare the effect of different initialization and column generation strategies on the performance of the algorithm for solving the linear relaxation.

2 Problem Description

The Minimum Converter Wavelength Assignment Problem (MCWAP) is to assign wavelengths to the links of a given set of lightpaths such that the total number of wavelength converters to be installed in the nodes of the network is minimized. For each lightpath, the routing path is already specified. Since the specific fiber and WDM system used by a lightpath first become relevant in the wavelength assignment, this decision is part of MCWAP. Hence, for every link, only the number and type of fibers and WDM systems is given, but lightpaths are not dedicated to specific equipment. We assume that both lightpaths and the installed equipment are bidirectionally oriented, i.e., fibers and WDM systems are installed in pairs, one for each direction, and a lightpath provides a virtual connection in both directions. The network is dimensioned in such a way that the number of channels consumed by all lightpaths on a link does not exceed the installed channel capacity. Likewise, we assume that the OXCs in the nodes are large enough (in terms of switching capacity) to handle all lightpaths.

From a graph theoretical point of view, MCWAP can be described as follows. Let $\mathcal{N} = (N, L)$ be an undirected graph with N representing the nodes and L the links of the physical topology. The set Λ contains all wavelengths of the available spectrum. At each link $\ell \in L$, κ_ℓ^λ defines the number of times wavelength $\lambda \in \Lambda$ is available by the installed equipment. Note that different WDM systems can result in different values κ_ℓ^λ for the diverse wavelengths. In case $\kappa_\ell^{\lambda_1} = \kappa_\ell^{\lambda_2}$ for all $\lambda_1, \lambda_2 \in \Lambda$, we use κ_ℓ as notation of the multiplicity of any wavelength on link $\ell \in L$. In total $K_\ell = \sum_{\lambda \in \Lambda} \kappa_\ell^\lambda$ optical channels are available on link $\ell \in L$.

As multiple lightpaths can be routed along the same path, we consider them cumulative. For a path p in $\mathcal{N}$, we denote with d_p the number of lightpaths to be established along this path. Let $N(p) \subset N$ denote the intermediate nodes along the path p (excluding its end nodes) while $L(p) \subset L$ denotes all links along the path p. All lightpaths that have to be assigned wavelengths are gathered in a multi-set $\mathcal{P}$ where each path p is contained d_p times (a multi-set is a set where element repetition is allowed). To differentiate the paths without multiplicity, the set $\mathcal{P}_1$ contains each path just once.

By the assumption that enough transmission capacity is available to establish all connections, lightpaths of length one can always be assigned a wavelength independently from the other lightpaths. Therefore, such lightpaths can be handled in a post-processing step and are left out in our further considerations.

3 Mathematical Formulations

In this section we derive and compare two integer linear programming formulations for MCWAP. The first formulation can be seen as the straightforward assignment of wavelengths. The second formulation is inspired by a column generation approach for the vertex coloring problem by Mehrotra and Trick [10].

3.1 Assignment Formulation

The standard way to formulate coloring-like problems is by the introduction of decision variables that represent the color assignments. As the assigned wavelength can differ for every link of a lightpath, we need variables $w_{p\ell}^\lambda$ to denote the number of times wavelength $\lambda \in \Lambda$ is assigned to the lightpaths established along path $p \in \mathcal{P}_1$ across link $\ell \in L(p)$. Note that these variables are integer in general as multiple lightpaths can be routed along path p and multiple WDM systems can be installed on a link.

To calculate the number of needed converters, a second set of variables z_{pv}^λ is necessary. They denote the number of converters needed in intermediate node $v \in N(p)$ to convert lightpaths routed along $p \in \mathcal{P}_1$ that are incoming in v at wavelength $\lambda \in \Lambda$. Now, MCWAP reads:

$$z_A = \min \sum_{p \in \mathcal{P}_1} \sum_{v \in N(p)} \sum_{\lambda \in \Lambda} z_{pv}^\lambda \tag{1}$$

$$\text{s.t.} \quad \sum_{\lambda \in \Lambda} w_{p\ell}^\lambda = d_p \qquad \forall p \in \mathcal{P}_1, \ell \in L \tag{2}$$

$$\sum_{p \in \mathcal{P}_1} w_{p\ell}^{\lambda} \leq \kappa_{\ell}^{\lambda} \qquad \forall \lambda \in \Lambda, \ell \in L \tag{3}$$

$$w_{puv}^{\lambda} - w_{pvw}^{\lambda} \leq z_{pv}^{\lambda} \qquad \forall p \in \mathcal{P}_1, v \in N(p), uv, vw \in L(p), \lambda \in \Lambda \tag{4}$$

$$w_{p\ell}^{\lambda}, z_{pv}^{\lambda} \in \mathbb{Z}_0^+ \tag{5}$$

The constraints (2) model that d_p wavelengths have to be assigned to each link $\ell \in L(p)$ of the path p. Inequalities (3) limit the use of $\lambda \in \Lambda$ on $\ell \in L$ to its availability κ_{ℓ}^{λ}.

By assigning values to the variables $w_{p\ell}^{\lambda}$ such that constraints (2) and (3) are satisfied, the wavelength assignment is specified. The converter counting requires to look at every path p and every non-end node v. For modeling purposes only, direct the path in an arbitrary way. Let uv and vw be respectively two consecutive links along p. If $w_{puv}^{\lambda} = w_{pvw}^{\lambda}$ for some wavelength $\lambda \in \Lambda$, then none of the corresponding lightpaths needs a converter. If $w_{puv}^{\lambda} \neq w_{pvw}^{\lambda}$, then the difference determines exactly the number of lightpaths that change from wavelength λ to another wavelength (in case $w_{puv}^{\lambda} > w_{pvw}^{\lambda}$) or from another wavelength to λ (in case $w_{puv}^{\lambda} < w_{pvw}^{\lambda}$). Summing up all such differences over all wavelengths in a node v clearly yields always zero, while the actual number of needed converters is given by accumulating only the positive differences. These differences are determined by inequalities (4) and summed up in the objective (1). Finally, fractional assignments are prohibited by the integrality constraints (5).

For uniform wavelength capacities, a major disadvantage of the assignment formulation is the degeneracy of solutions caused by the symmetry of the spectrum Λ. Not only for all integer solutions, but also for fractional solutions of linear relaxations (with/without cutting planes), $\mathcal{O}(|\Lambda|!)$ equivalent solutions exist. For problems with such characteristics like vertex coloring [11] and frequency assignment [1], cutting plane approaches and branch-and-bound based on similar assignment formulations have been shown to be computationally intractable. This motivates to investigate other formulations for the problem at hand.

3.2 Path Packing Formulation

To overcome the color symmetry degeneracy for the vertex coloring problem, Mehrotra and Trick [10] introduced a column generation approach. Given a graph G, it is well known that all vertices that can be colored by the same color form a stable set. Thus the chromatic number $\chi(G)$ is given by the minimum number of stable sets needed to cover all vertices. By introducing a variable for every stable set in the graph, an alternative formulation for vertex coloring is derived. This successful approach has inspired to derive a similar formulation for MCWAP. A step-by-step generalization of the formulation for vertex coloring to MCWAP can be found in [7]. To simplify the explanation, we assume uniform wavelength capacities in the sequel.

The key to our approach is the observation that consecutive links of a lightpath with the same wavelength can be gathered to a subpath. In this view, a lightpath consists of a number of subsequent subpaths (possibly only one) that cover the path. All subpaths that are assigned the same wavelength within the network can be viewed as a *packing of subpaths*. Now, any feasible wavelength assignment decomposes into at most $|\Lambda|$ subpath packings.

Furthermore, the number of converters needed for a lightpath is exactly the number of subpaths to cover the lightpath minus one. Since the total number of lightpaths is fixed, the objective of MCWAP is equivalent to minimizing the total number of subpaths involved.

To be able to formulate the above problem statement as an integer program, we have to characterize the feasible packings of subpaths. We introduce the following notation. For each $p \in \mathcal{P}_1$, let $\mathcal{S}_p$ denote the set of all subpaths s of p. Note that the same subpath can be in multiple sets $\mathcal{S}_p$ and $|\mathcal{S}_p| = \frac{1}{2}|L(p)|(|L(p)|+1)$. Let $\mathcal{S} = \cup_{p\in\mathcal{P}_1}\mathcal{S}_p$ denote the set of all possible subpaths. Then a *path packing* ϕ is a multi-set of items of $\mathcal{S}$ such that all subpaths $s \in \phi$ can be assigned the same wavelength λ, i.e., for every link $\ell \in L$, at most κ_ℓ^λ subpaths containing link ℓ are in the set ϕ. The multiplicity of each subpath $s \in \mathcal{S}$ in the path packing ϕ is denoted by t_ϕ^s. The collection of all multi-sets of $\mathcal{S}$ that are path packings is denoted by Φ.

A path packing $\phi \in \Phi$ can be selected more than once, if the multi-set of subpaths to be assigned the same wavelength is identical for multiple wavelengths. For every path packing $\phi \in \Phi$, we therefore introduce a general integer variable x_ϕ denoting the number of times all subpaths $s \in \phi$ are assigned the same wavelength. To specify the subpaths that are used to cover a path $p \in \mathcal{P}_1$, we introduce a second class of variables y_p^s which denote the number of times subpath s is used to cover the lightpaths routed along path $p \in \mathcal{P}_1$. Now, MCWAP reads alternatively:

$$z_P = \min \sum_{p\in\mathcal{P}_1}\sum_{s\in\mathcal{S}_p} y_p^s \qquad \left[-\sum_{p\in\mathcal{P}_1} d_p\right] \tag{6}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}_p:\ell\in L(s)} y_p^s = d_p \qquad \forall p \in \mathcal{P}_1, \ell \in L(p) \tag{7}$$

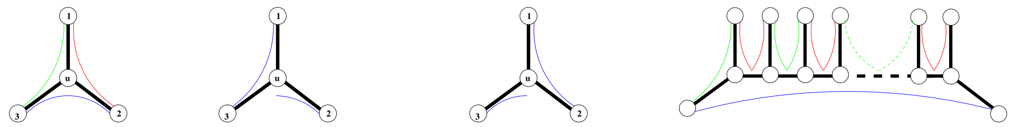
$$\sum_{\phi\in\Phi} t_\phi^s x_\phi = \sum_{p\in\mathcal{P}_1:s\in\mathcal{S}_p} y_p^s \qquad \forall s \in \mathcal{S} \tag{8}$$

$$\sum_{\phi\in\Phi} x_\phi \le |\Lambda| \tag{9}$$

$$y_p^s, x_\phi \in \mathbb{Z}_0^+ \tag{10}$$

As expressed by (6), the number of converters is given by the number of subpaths minus the total demand sum, the latter stating a fixed value. At every link $\ell \in L$, the lightpath multiplicity for each path $p \in \mathcal{P}_1$ has to be satisfied, which is enforced by constraints (7). Constraints (8) model that every subpath $s \in \mathcal{S}$ is offered by the selected path packings as often as chosen for covering lightpaths. Finally, constraint (9) restricts the number of selectable path packings to the size of the available spectrum Λ, and constraints (10) guarantee integrality.

For non-uniform wavelength spectra, this formulation can be adapted by defining a separate set of path packing variables for every subset of wavelengths Λ^k for which $\kappa_\ell^{\lambda_1} = \kappa_\ell^{\lambda_2}$ for all $\lambda_1, \lambda_2 \in \Lambda^k$. Typically the number of subsets is limited by the restricted number of different WDM system types that are installed.



(a) Instance

(b) Two path packings with $x_\phi = 1$ in the linear relaxation of (6)–(10)

(c) Extended network with arbitrarily large difference between the LP values

Figure 1: Star network with different linear relaxation values

3.3 Comparison

Let z_A^* denote the optimal value of the linear relaxation of the assignment formulation (1)–(5). In case of uniform wavelength capacities, the following observation can be made:

Lemma 1 *If $\kappa_\ell^\lambda = \kappa_\ell$ for all $\lambda \in \Lambda$, $\ell \in L$, then $z_A^* = 0$.*

Proof: Set all variables $w_{p\ell}^\lambda = d_p/|\Lambda|$ and use that $\kappa_\ell = K_\ell/|\Lambda|$. □

Let z_P^* denote the value of the linear relaxation of the path packing formulation (6)–(10). It is easy to verify that every (fractional) solution of (6)–(10) can be transformed to a (fractional) solution of (1)–(5), which yields:

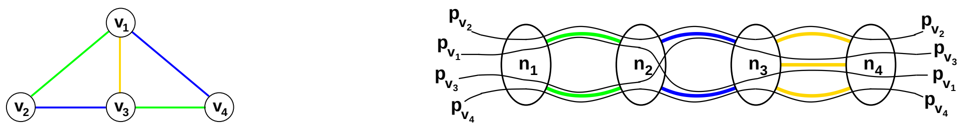
Lemma 2 *The objective value of the linear relaxation of (6)–(10) is not worse than the objective value of the linear relaxation of (1)–(5), i.e., $z_P^* \geq z_A^*$.*

Equivalence of both formulations does not hold, which can be shown by the instance displayed in Figure 1(a) with $d_p = 1$ for all paths. In case $|\Lambda| = 2$ and $\kappa_\ell = 1$ for all links, one wavelength converter is clearly needed. The linear relaxation of (1)–(5) has the value zero, whereas the linear relaxation of (6)–(10) has the value one since every path packing covering all links by subpaths contains single links as subpaths, cf. Figure 1(b). In fact, by extending the star network as displayed in Figure 1(c), the following result is obtained:

Lemma 3 *The difference $z_P^* - z_A^*$ can be arbitrarily large, even if only two wavelengths are involved.*

For $|\Lambda| > 2$, the example of Figure 1(a) can be generalized by setting $d_p = \frac{1}{2}|\Lambda|$ for each of the three lightpath requests. Then $z_P = z_P^* = \frac{1}{2}|\Lambda|$, while $z_A^* = 0$ still.

Computational experiments (cf. Section 5) show that in those cases where we know the optimal solution, the value of the linear relaxation of (6)–(10) equals the optimal value. The suggestion that this is always the case is however not true. To see this, we carry over some results for vertex coloring to wavelength assignment. It is well known that in case $\kappa_\ell = 1$ for all links $\ell \in L$, the question whether there exists a converter-free wavelength assignment is equivalent to the question whether there exists a vertex coloring with at most $|\Lambda|$ colors



(a) Graph G (b) Resulting network $\mathcal{N}_G$ with paths p_v

Figure 2: Example of the construction of a wavelength assignment instance from a vertex coloring instance by help of edge coloring

in the so-called *path conflict graph*. In the path conflict graph $G_{\mathcal{P}}$, we have a vertex v_p for every lightpath $p \in \mathcal{P}$ and connect two vertices by an edge if the corresponding lightpaths share at least one link. As we will show, any graph can be obtained by this construction as conflict graph $G_{\mathcal{P}}$ of a MCWAP instance, which directly leads to the $\mathcal{NP}$-completeness of wavelength assignment.

Given a graph $G = (V, E)$, there are several ways to construct a wavelength assignment problem with G as its path conflict graph. One such a reverse construction is as follows: Color the edges of G such that no two edges incident to a vertex obtain the same color. The minimum number of colors needed for such an edge coloring is the chromatic index $\chi'(G)$, but also a coloring with more colors will do (e.g., for simple graphs a coloring with $\Delta(G) + 1$ can be computed in polynomial time [12]). Let c be the number of colors used, and let $E_1, \ldots, E_c \subset E$ denote the edge color classes. Now, we construct an optical network $\mathcal{N}_G = (N, L)$ with $|N| = c{+}1$ nodes. For $i = 1, \ldots, c$, the nodes n_i and n_{i+1} are connected by multiple (parallel) links (if no parallel links are allowed, this can be avoided by introducing an intermediate node on each link). For each vertex $v \in V(G)$, we define a path p_v with $d_p = 1$ from node n_1 to n_{c+1} traversing all intermediate nodes. The number of paths on a link is either one or two. The paths p_v and p_w share a link between n_i and n_{i+1} if and only if edge $vw \in E_i$. If $v \in V(G)$ is not incident to any edge within color class i, then path p_v uses a separate link between n_i and n_{i+1}. Thus, the total number of parallel links between n_i and n_{i+1} is $|E_i| + |V| - 2|E_i| = |V| - |E_i|$. See Figure 2 for an example. From the construction, the following lemma follows directly:

Lemma 4 *Let $G = (V, E)$ be a graph. Then there exists a wavelength assignment instance $(\mathcal{N}_G, \mathcal{P}, \Lambda)$ with $\kappa_\ell = 1$ for all $\ell \in L$, such that no conversion is needed if and only if $\chi(G) \leq |\Lambda|$.*

Recall that the chromatic number $\chi(G)$ of a graph G is the minimum number of stable sets such that all vertices are covered. If we allow to take stable sets in fractional amounts under the restriction that the sum of the fractions still covers all vertices, we obtain the *fractional chromatic number* $\chi^*(G)$. In terms of integer programming, the fractional chromatic number is the value of the linear relaxation of the stable set formulation for vertex coloring.

Theorem 1 *Let $G = (V, E)$ be a graph with $\chi^*(G) \leq \chi(G) - 1$ and $(\mathcal{N}_G, \mathcal{P}, \Lambda)$ define the constructed wavelength assignment instance with $\kappa_\ell = 1$ for all $\ell \in L$. If $|\Lambda| = \chi(G) - 1$, then $z_P^* = 0$, whereas $z_P > 0$.*

Proof: The fractional chromatic number $\chi^*(G)$ is attained by a collection $\mathcal{B}$ of stable sets $B \subseteq V$ in G, each one taken with a fractional value $0 < w_B \leq 1$. We identify a path packing in ϕ_B for each $B \in \mathcal{B}$. Let $t^s_{\phi_B} = 1$ if subpath $s = p_v$ for some $v \in B$, and $t^s_{\phi_B} = 0$ otherwise. Now, set $x_{\phi_B} = w_B$ for all $B \in \mathcal{B}$ and $y^s_p = 1$ if $s = p$, otherwise $y^s_p = 0$. Then constraints (7) and (8) are satisfied by construction, and constraint (9) is satisfied by $\sum_{B\in\mathcal{B}} x_{\phi_B} = \sum_{B\in\mathcal{B}} w_B = \chi^*(G) \leq \chi(G) - 1$. The value of this fractional solution is zero, hence $z^*_P = 0$, while $z_P > 0$ due to Lemma 4. $\square$

It is easy to find graphs for which $\chi^*(G) < \chi(G)$, e.g., for an odd cycle C_{2k+1}, $k > 0$, $\chi^*(C_{2k+1}) = 2 + 1/k$, whereas $\chi(C_{2k+1}) = 3$. Thus, $\chi^*(C_{2k+1}) < \chi(C_{2k+1})$ for $k \geq 2$. Also the Petersen graph has $2.5 = \chi^*(G) < \chi(G) = 3$. However, the gap $\chi(G) - \chi^*(G)$ for all these graphs is smaller than one, and thus we cannot apply Theorem 1.

Kneser graphs are a generalization of the Petersen graph. Given two positive integers n and k, the Kneser graph $KG_{n,k}$ is the graph whose vertices represent the k-subsets of $\{1, \dots, n\}$, and where two vertices are connected if and only if they correspond to disjoint subsets [9]. $KG_{n,k}$ has $\binom{n}{k}$ vertices, each one with degree $\binom{n-k}{k}$ (we assume $\binom{n-k}{k} \equiv 0$ if $n - k < k$). The Petersen graph equals to the case $n = 5$, $k = 2$. For $n = 3k - 1$, it holds that $\chi(KG_{3k-1,k}) = k + 1$, whereas $\chi^*(KG_{3k-1,k}) = \frac{3k-1}{k} < 3$. Thus for $k \geq 3$, $\chi^*(KG_{3k-1,k}) \leq \chi(KG_{3k-1,k}) - 1$, and hence we can construct an optical network with the property of Theorem 1. In particular, for $KG_{8,3}$ there exists an edge coloring with $\Delta(G) + 1 = 11$ colors, yielding an optical network with 12 nodes and 336 links for which $z^*_P = 0$ whereas $z_P > 0$.

4 Column Generation Algorithm

The number of columns of the formulation (6)–(10) is tremendously large due to the exponential number of path packings. A size reduction is possible by relaxing constraints (8) to

$$\sum_{\phi\in\Phi} t^s_\phi x_\phi \geq \sum_{p\in\mathcal{P}_1 : s\in\mathcal{S}_p} y^s_p \qquad \forall s \in \mathcal{S} \tag{11}$$

which allows a subpath to be covered more often by the path packings than the desired number. In this case, we can restrict to maximal path packings—a significantly smaller subset of Φ.

Despite the substantial size reduction, the number of columns is still exponentially and hence too large to be considered explicitly in practice. Therefore, like in Mehrotra and Trick [10], we propose a column generation approach, where only a subset of the columns is stored explicitly. After computation of the linear relaxation for this subset, other profitable columns are generated with a so-called pricing problem. In the pricing problem, a profitable column is selected based on the values of the dual variables. In case no profitable columns can be generated anymore, the linear relaxation including all columns has been solved (cf. [3, 4] for further details). The algorithm can be incorporated within a branch-and-bound (or branch-and-cut) scheme, resulting in a branch-and-price (or branch-cut-and-price) algorithm.

For MCWAP, we apply column generation for the x variables, whereas all y variables are considered explicitly. Let $\overline{\Phi} \subset \Phi$ denote the actual subset of path packings included. To formulate the pricing problem for MCWAP, we introduce the dual variables π_ℓ^p, π^s, and π^Λ for the constraints (7), (8), and (9), respectively. From linear programming, we know that a primal-dual pair $((x, y), \pi)$ is optimal for the linear programming relaxation of (6)–(10), whenever $c - A^T\pi \leq 0$, with c the primal objective function, and A the coefficient matrix. For a path packing $\phi \in \Phi$, $c_\phi = 0$, and the coefficients of A corresponding to (7) equal zero as well. So, the optimality condition transforms to

$$-\sum_{s\in\mathcal{S}} t_\phi^s \pi^s \leq \pi^\Lambda \tag{12}$$

Note that $\pi^s \leq 0$ by (11), whereas $\pi^\Lambda \geq 0$. If optimality is reached, (12) holds for every $\phi \in \Phi$. To verify this, we search for the path packing $\phi \in \Phi$ that maximizes the left hand side of (12). If the maximum is less than or equal to π^Λ, then no improving columns exist anymore, and the linear relaxation is solved. Otherwise, we have found a path packing $\phi \in \Phi$ that violates (12) and can be added to the linear program to improve the relaxation.

Maximizing $-\sum_{s\in\mathcal{S}} \pi^s t_\phi^s$ can be formulated as an optimization problem. We introduce the integer variables t^s for all $s \in \mathcal{S}$ representing the multiplicity function of a path packing. Then the pricing problem reads

$$z = \max \sum_{s\in\mathcal{S}} -\pi^s t^s \tag{13}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}:\ell\in L(p)} t^s \leq \kappa_\ell \qquad \forall \ell \in L \tag{14}$$

$$t^s \in \mathbb{Z}_0^+ \qquad \forall s \in \mathcal{S} \tag{15}$$

In case $\kappa_\ell = 1$ for all $\ell \in L$, the pricing problem reduces to a maximum weighted set packing. Hence, the pricing problem is $\mathcal{NP}$-hard in general. However, the pricing problem needs not to be solved to optimality in every iteration of the algorithm. It suffices to find a solution with value strictly larger than π^Λ, e.g., by heuristics. Only if this fails, we have to solve (13)–(15) to optimality.

5 Computational Experiments

Since the linear relaxation value of the assignment formulation always equals zero, a computational evaluation is of no worth and thus is left out. To solve the master problem (6)–(10) and the pricing problem (13)–(15), we follow the general outline of a column generation algorithm [4]. However, several strategies can be applied to obtain the linear relaxation value as fast as possible. After explaining the instance generation, we compare in Section 5.2 strategies for the initialization of the algorithm with a small subset of all possible path packings, whereas Section 5.3 is devoted to strategies to generate new columns. Finally, we discuss the results.

network	$\|N\|$	$\|L\|$	$\|\mathcal{P}\|$ for survivability fraction $p=0$ "up"	$p=\frac{1}{3}$ "low"	$p=\frac{2}{3}$ "high"	$p=1$ "fp"	total # instances	# non-zero best sol.
US	14	21	2710	2710	3166	4655	20	0
Germany	17	26	686	699	836	1193	20	2
Germany ext.	17	28	686	699	796	1122	20	5
Europe	28	41	1008	1148	1480	1855	20	16

(a) network characteristics

network	average $\|\mathcal{P}_1\|$ $p=0$ "up"	$p=\frac{1}{3}$ "low"	$p=\frac{2}{3}$ "high"	$p=1$ "fp"
Germany	-	-	75	64
Germany ext.	46.3	82	-	102
Europe	352	752	829.4	840.2

(b) average number of different paths for non-zero instances

Table 1: Instance characteristics ("up" stands for unprotected, "fp" for full protected)

5.1 Instance Generation

The algorithm is tested on a number of instances, derived from three reference scenarios defined within the MultiTeraNet project [2]. Every scenario consists of a network topology and a traffic matrix which specifies the demand for each pair of nodes as number of lightpaths to establish. The networks represent an US network based on the NSF topology, a hypothetical German network, and an (also hypothetical) European network. A fourth scenario has been created by adding two links to the German network as to increase the connectivity, cf. [8]. For each scenario, we consider four survivability specifications which differ in the fraction $p \in [0,1]$ of the traffic that has to be protected against any single link or node failure. We use $p = k \cdot \frac{1}{3}$ for $k = 0,1,2,3$ as protection levels. For a demand of d lightpaths, $\lceil p \cdot d \rceil$ of them have to be protected by additional establishment of backup lightpaths according to the concept Demand-wise Shared Protection proposed in [8]. In Table 1, a summary of the instance characteristics is given. With the network optimization engine OND (Optical Network Design, cf. [13]), a dimensioning and routing can be computed, given installable devices like fibers, WDM systems, and OXCs. In fact, multiple configurations are generated during the optimization. For each scenario, we have extracted the five best configurations generated within one hour of computation time. These configurations form the input for the wavelength assignment subproblem. In all instances, the fibers are equipped with WDM systems providing $|\Lambda| = 40$ wavelengths, while technical properties of fibers and OXCs are unimportant for studying MCWAP.

An upper bound on the number of wavelength converters for each of these 80 instances is computed by the iterative improvement heuristics as described in [6] (with a maximum of 6000 CPU seconds running time). The number of non-zero solutions is reported in the last column of Table 1. From these values, it can be concluded that in many cases a zero converter solution could be found, and hence $0 \leq z_P^* \leq z_P = 0$. The input of our computational study consists of the remaining instances, which potentially have a non-zero solution.

The column generation algorithm has been implemented in C++, using CPLEX 9.0 [5] as (integer) linear programming solver with ILOG's Concert Technology as interface. All computations have been carried out on a Linux-operated PC with a 3.2 GHz Pentium IV processor.

5.2 Initialization Strategies

In every iteration of the column generation algorithm, the set of path packings considered explicitly is extended. It is however unclear with which columns the algorithm should be initialized as to minimize the number of iterations (and computation time). In this section, we describe three strategies that turned out to be most promising in our experiments.

The path packing formulation with (11) instead of (8) allows to initialize the master problem with a single path packing: For every link, we can take the subpath consisting only of this single link κ_ℓ times. Obviously this path packing is feasible. We refer to this path packing as the *basic column*. This column can be taken $|\Lambda|$ times to obtain a feasible solution in which every lightpath is converted at every intermediate node. Hence, the solution value is the worst possible one as every path p contributes $(|L(p)| - 1)d_p$ to the objective.

Alternatively, the algorithm can be initialized with a feasible wavelength assignment obtained by any heuristic method. Given such a solution, we have to construct $|\Lambda|$ path packings consisting of all subpaths to which the same wavelength is assigned. In the strategy *best solution*, we initialize the algorithm with the best known solution (cf. Section 5.1) in addition to the basic column.

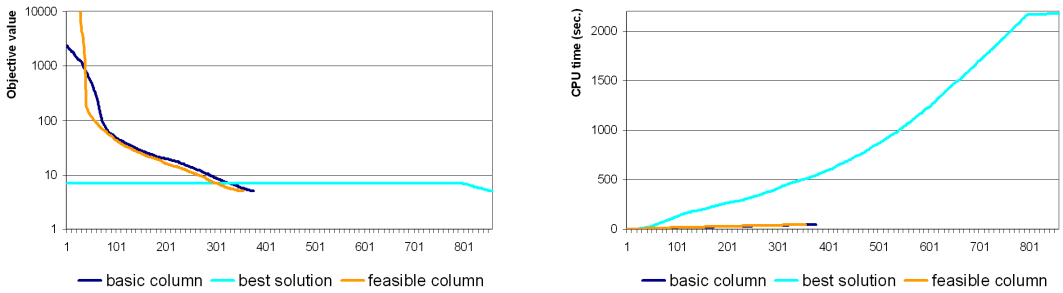
Both strategies described above start with a feasible solution consisting of a number of path packings. A drawback of such an initialization is the necessity to find new columns that can be combined with the already available columns. To gradually construct a feasible (fractional) solution from scratch, the algorithm has to be initialized without any path packing. However, in that case the first master problem is infeasible and typically no dual information is generated by the linear programming solver, which makes it impossible to solve the first pricing problem. A remedy lies in the usage of the basic column in a different way. We add a column to the formulation that is equivalent to the basic column, except for the objective coefficient and the contribution to the spectrum bound inequality (9). This so-called *feasibility column* does not contribute to the left hand side of (9) and has a huge objective penalty. It guarantees that the first master problem is feasible by taking the column (at most) $|\Lambda|$ times. Hence, the first LP value will be very high. In the next iterations, columns are generated that replace the feasibility column more and more until a (fractional) solution is found making the feasibility column obsolete. This strategy is referred to as *feasibility column*.

instance	best sol.	z_P^*	basic col. # col.	basic col. time	best sol. # col.	best sol. time	feas. col. # col.	feas. col. time
europe-up1	7	2	447	66	560	1311	393	54
europe-up2	2	0	352	42	989	3828	334	41
europe-up3	7	5	377	49	859	2181	356	47
europe-up4	5	1	357	48	826	3022	350	50
europe-up5	1	0	344	45	821	2480	347	44
europe-low3	7	0	762	1270	870	19492	725	907
europe-high1	56	0	1558	4290	1377	6260	1228	3062
europe-high2	57	0	1505	4149	1441	7820	1365	3549
europe-high3	31	0	1415	4039	1183	11364	1220	2808
europe-high4	57	0	1591	5337	1360	8152	1322	6180
europe-high5	5	0	1311	3268	1650	124319	998	3355
europe-fp1	13	0	1669	5499	1502	49713	1287	3409
europe-fp2	20	0	1591	4999	1758	35841	1212	4483
europe-fp3	19	0	1555	4813	1445	37878	1193	4193
europe-fp4	29	0	1543	4537	1304	22967	1141	4054
europe-fp5	42	0	1595	4965	1504	13060	1265	4541
germany-high2	4	4	44	0	2	0	42	0
germany-fp1	8	8	34	0	0	0	39	0
germany+-up2	16	16	37	0	2	0	47	0
germany+-up3	12	12	34	0	4	0	49	0
germany+-up5	9	9	24	0	2	0	28	0
germany+-low2	9	9	32	0	5	0	48	0
germany+-fp5	5	5	94	1	3	0	97	2

Table 2: Effect of different initialization strategies on the performance of the column generation algorithm (CPU times in seconds)

Compared with the basic column strategy, the feasibility column strategy progresses differently since (9) is not satisfied with equality in the feasibility column case, whereas it is in the basic column case. Hence, the dual variables have different values and thus likely produce different columns to add.

The three different strategies are tested for all instances with a non-zero best solution. Table 2 shows the results. Besides the value of the best solution and z_P^*, we present for each of the strategies the number of columns generated (# col.) and the CPU time in seconds (time). For the instance **europe-up3**, the progress with the number of generated columns of the LP value (with logarithmic scale) and the accumulating CPU time are displayed in Figure 3. A discussion of the results is postponed till Section 5.4.



(a) LP value

(b) CPU time

Figure 3: Iterative progress of LP value (in logarithmic scale) and accumulated CPU time for instance **europe-up3**

5.3 Pricing Strategies

As already pointed out in Section 4, the pricing problem has not to be solved to optimality at each iteration of the algorithm. Only if heuristics fail to find a path packing with value strictly larger than π^Λ, we have to solve (13)–(15) to optimality. In our computations, the CPLEX integer programming solver is used as an heuristic by limiting the maximum number of branch-and-bound nodes to 100. Only in case the best solution found after reaching this limit has value $z \leq \pi^\Lambda$, we continue the branch-and-bound until a solution is found with value $z > \pi^\Lambda$ or until it has been proven that no such solution exists. In the latter case, the linear programming relaxation is solved to optimality.

Note that in case the pricing problem is solved in less than 100 nodes of the branch-and-bound tree, the heuristic is exact. In fact, our experiments show that typically the more the algorithm advances, the more pricing problems are solved to optimality within these 100 branch-and-bound nodes, from almost never in the first iterations till almost always in the last iterations.

One other aspect of the pricing problem on which we can take influence is the multiplicity of the subpaths in a path packing. In addition to the constraints (14) and (15), we can restrict the path packings by the following argument. For a subpath s, it does not make sense to generate path packings with $t^s > D_s$, where $D_s := \sum_{p \in \mathcal{P}_1 : s \in \mathcal{S}_p} d_p$ is the maximum number of times that the subpath can be used in a wavelength assignment. So, we can bound the variables t^s from above by D_s. This has been carried out in all our computations.

Furthermore, although D_s states a senseful upper bound on the number of times a subpath can be taken in any path packing, exceeding d_s (where $d_s = 0$ if $s \notin \mathcal{P}_1$) directly implies the placement of converters on some lightpaths. As typically the number of converters is small in comparison with the number of lightpaths and the lightpaths are partitioned among $|\Lambda|$ wavelengths, the number of path packings in a solution with some t^s exceeding d_s will be small. In the strategy *restricted pricing*, we therefore start with the generation of path packings for which the variables t^s are bounded by d_s, instead of D_s. As soon as no further

	with restricted pricing				reduction in %			
	basic col.		feas. col.		basic col.		feas. col.	
instance	# col.	time	# col.	time	# col.	time	# col.	time
europe-up1	252	24	298	56	44%	64%	24%	-4%
europe-up2	157	15	210	32	55%	64%	37%	22%
europe-up3	199	17	290	48	47%	65%	19%	-2%
europe-up4	239	21	269	45	33%	56%	23%	10%
europe-up5	168	18	282	47	51%	60%	19%	-7%
europe-low3	356	346	326	667	53%	73%	55%	26%
europe-high1	570	1158	563	1868	63%	73%	54%	39%
europe-high2	558	1073	573	1884	63%	74%	58%	47%
europe-high3	453	754	478	1504	68%	81%	61%	46%
europe-high4	482	929	582	2049	70%	83%	56%	67%
europe-high5	388	583	392	1463	70%	82%	61%	56%
europe-fp1	534	967	564	1279	68%	82%	56%	62%
europe-fp2	699	1585	458	1992	56%	68%	62%	56%
europe-fp3	481	891	632	2666	69%	81%	47%	36%
europe-fp4	448	794	504	1870	71%	82%	56%	54%
europe-fp5	495	979	629	2600	69%	80%	50%	43%

Table 3: Effect of the restricted pricing strategy on the performance of the column generation algorithm (CPU times in seconds), compared to the default pricing strategy

restricted path packings can be generated, the upper bound on the variables in the pricing problem is lifted from d_s to D_s.

Table 3 contains the results in number of generated columns and CPU time with restricted pricing as well as the reduction compared to the default pricing strategy. We left out the German instances since the values without restricted pricing (cf. Table 2) are too small for a reasonable comparison. Moreover, we only compare the two most promising strategies for initialization: basic column and feasibility column.

5.4 Discussion

The first thing that can be observed from the results in Table 2 is the unregular behavior of the best solution strategy. For the German instances, the computation times and the number of columns generated can be neglected; for the European instances, the strategy is outperformed by the other strategies. In particular, the CPU times are not comparable. This effect can be explained easily. In those cases that the LP value equals the best known solution value, there cannot exist columns that indeed decrease the value of the initial LP. This conclusion is drawn by the (best solution strategy) algorithm after a few iterations. A direct consequence of this is that we know the optimal value for these instances. If in contrast the best solution value is much higher than the optimal LP value, starting with such a solution, being forced to improve it step-wise, hampers the progress, compared to the other strategies which can more simple and direct construct a better LP solution.

A closer look at the values discloses that those instances with a small gap between LP and best solution value are in fact the most time consuming ones in comparison with the other strategies. This second effect is more difficult to explain, since it is in fact contra-intuitive. Experiments with less good solutions used as initial columns revealed that they often result in better CPU times as the best solution. A possible explanation is that the progression can be more easily achieved if the solution is not that close to the LP value. Figure 3(b) confirms this, as the slope of the CPU time is far more steep for the best solution strategy than for the other strategies, indicating that it is more difficult to integrate the new columns (the CPU time for pricing is typically only marginal in comparison with the reoptimization of the master LP). First if the LP value starts to drop (cf. Figure 3(a)), the CPU time per column is comparable to the other strategies, but this usually happens much later.

From the results in Table 2, the feasibility column strategy seems to be most favorable, closely followed by the basic column strategy. Figure 3(a) shows that after a different start both strategies follow the same curve.

Using restricted pricing turns out to be a very profitable idea for solving the linear relaxation more quickly. Reductions in number of columns above 50% are typical, CPU time reductions of more than 80% not seldom. The best results are now obtained by the basic column strategy. In case $z_P^* = 0$, no switch to unrestricted pricing is required at all. In case $z_P^* > 0$, a change to unrestricted pricing is at some point necessary. For the European instances, the optimal LP value is at this point already reached and less than ten unrestricted columns are needed to prove optimality. For the German instances, also only a small number of unrestricted columns is necessary, but the LP value is reduced further by these columns.

Although the computation times decrease significantly by restricted pricing, the times are still relatively large for solving a linear relaxation. This is reasoned by the LP size: for example, the instance **europe-fp2** has 16,102 y-variables and 5779 rows.

Altogether, we can conclude that the path packing formulation is favorable. The associated LP relaxation has often a non-zero optimum which states a lower bound and frequently directly proves optimality of the best known solution generated heuristically. Such a confirmation is typically obtained very quickly. In case the best known solution does not equal the LP value, we have some good strategies for fastly solving the LP. The computational comparison revealed that the most profitable strategy combination is an initialization with the basic column with restricted pricing as long as it provides potential columns. Unfortunately, not in all cases the gap between lower and upper bound is reduced by the algorithm (e.g., for the larger European instances). It is not clear in these cases whether the LP value is bad or simply the heuristics did not manage to find better solutions (we tend to the latter claim). In any case, the path packing formulation turns out to be a good instrument to investigate MCWAP.

6 Concluding Remarks

In this paper, we presented a column generation approach for the minimum converter wavelength assignment problem in optical networks. We have shown that the lower bound provided by the linear relaxation of the path packing formulation is both in theory and in practice significantly better than the one by a straightforward assignment formulation. In fact, for 7 out of 23 non-trivial instances, we could prove optimality of the best known solution by solving the linear relaxation of the path packing formulation.

A computational comparison of alternative initialization and column generation strategies for solving the linear relaxation of the path packing formulation revealed that such aspects have to be chosen carefully. This is of particular importance in the context of the design of a branch-and-price algorithm to find the optimal solution for MCWAP. Such an algorithm remains the major challenge to be solved by further research. Except for the initialization and column generation strategies, it is of equal importance to find good branching strategies.

By recently developed heuristic technique advancements [7], the best known solution could be improved to the value of the linear relaxation for two more instances, again proving optimality. Besides further founding the benefit of our approach, this observation also indicates that the heuristic algorithms are still improvable, stating another direction for further research.

References

[1] K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for the frequency assignment problem. *4OR*, 1(4):261–317, 2003.

[2] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer, J. Späth, C. Gauger, and M. Köhn. Reference transport network scenarios. Technical report, BMBF MultiTeraNet, July 2003. http://www.ikr.uni-stuttgart.de/IKRSimLib/Referenz_Netze_v14_full.pdf.

[3] V. Chvátal. *Linear Programming.* W.H. Freeman and Company, New York, 1983.

[4] J. Desrosiers and M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation.* Kluwer Academic Publishers, Boston, MA, 2005.

[5] ILOG. CPLEX version 9.0, 2003. http://www.ilog.com/products/cplex.

[6] A. M. C. A. Koster and A. Zymolka. Minimum converter wavelength assignment in all-optical networks. In *Proceedings of ONDM 2004*, pages 517–535, Ghent, Belgium, 2004. The 8th IFIP Working Conference on Optical Network Design & Modelling.

[7] A. M. C. A. Koster and A. Zymolka. Provably good solutions for wavelength assignment in optical networks. In *Proceedings of ONDM 2005*, Milan, Italy, 2005. The 9th IFIP Working Conference on Optical Network Design & Modelling.

[8] A. M. C. A. Koster, A. Zymolka, M. Jäger, and R. Hülsermann. Demand-wise shared protection for meshed optical networks. *Journal of Network and Systems Management*, 13(1), 2005.

[9] L. Lovász. Kneser's conjecture, chromatic numbers and homotopy. *Journal of Combinatorial Theory Series A*, 25:319–324, 1978.

[10] A. Mehrotra and M. A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8:344–354, 1996.

[11] P. Pardalos, T. Mavridou, and J. Xue. The graph coloring problem: A bibliographic survey. In D.-Z. Du and P. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 2, pages 331–395. Kluwer Academic Publishers, 1998.

[12] V. G. Vizing. On an estimate of the chromatic class of a p-graph (Russian). *Diskret. Analiz.*, 3:25–30, 1964.

[13] A. Zymolka, A. M. C. A. Koster, and R. Wessäly. Transparent optical network design with sparse wavelength conversion. In *Proceedings of ONDM 2003*, pages 61–80, Budapest, Hungary, 2003. The 7th IFIP Working Conference on Optical Network Design & Modelling.