Solving the Static Design Routing and Wavelength Assignment Problem

Helmut Simonis*

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
h.simonis@4c.ucc.ie

Abstract. In this paper we present a hybrid model for the static design variant of the routing and wavelength assignment problem in directed networks, an important benchmark problem in optical network design. Our solution uses a decomposition into a MIP model for the routing aspect, combined with a graph coloring step modelled using either MIP (Coin-OR), SAT (minisat) or finite domain constraints (ECLiPSe). We consider two possible objective functions, one minimizing the maximal number of frequencies used on any of the links, the other minimizing the total number of frequencies used. We compare the models on a set of benchmark tests, results show that the constraint model is much more scalable than the alternatives considered, and is the only one producing proven optimal or near optimal results when minimizing the total number of wavelengths.

1 Introduction

The routing and wavelength assignment problem (RWA) [10,1,17] in optical networks considers a network where demands can be transported on different optical wavelengths through the network. Each accepted demand is allocated a path from its source to its sink, as well as a specific wavelength. Demands routed over the same link must be allocated to different wavelengths, while demands whose paths are link disjoint may use the same wavelength.

The RWA problem is a well studied, important problem in optical network design, for which many problem variants have been considered. Depending on the technology used, the network may be assumed to be *directed* or *undirected*. The *static design problem* considers the problem of allocating all given demands on the network topology, using the minimal number of frequencies. The *demand acceptance problem* considers a fixed, given number of frequencies on all links in the network. The objective is to accept the maximal number of demands in the network. In this paper we discuss the static design problem in a directed network, while a constraint-based solution for the demand acceptance problem has been described in [13].

More formally, we are considering a directed network $G = (N, E)$ of nodes N and edges E. A demand $d \in D$ is between source $s(d)$ and sink $t(d)$. We use the notation

* This work was supported by Science Foundation Ireland (Grant Number 05/IN/I886). Support from Cisco Systems and the Silicon Valley Community Foundation is gratefully acknowledged.

J. Larrosa and B. O'Sullivan (Eds.): CSCLP 2009, LNAI 6384, pp. 59–75, 2011.

In(n) and Out(n) to denote all edges entering resp. leaving node n. An a priori upper bound on the number of available wavelengths is required, we use the set Λ for this purpose.

Figure 1 shows one of the example networks we will use in the evaluation, with just two demands (5-13) and (1-12). On the left, the demands are allocated to different frequencies (colours), and thus can share link 8-9, on the right they use the same frequency, and therefore must be routed on link disjoint paths.

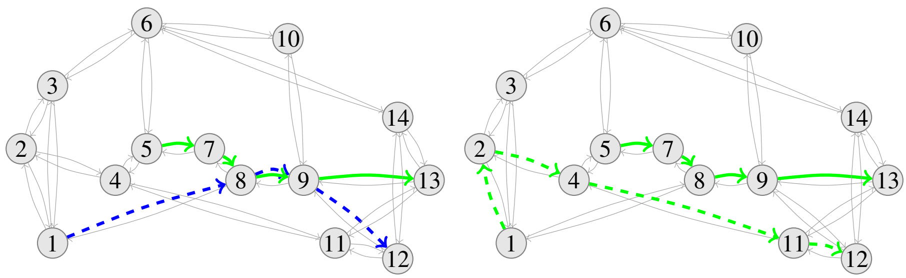

Fig. 1. Example Network `nsf` with 2 Demands

1.1 Basic Problem

We can formulate a *basic model* of the problem with two sets of 0/1 integer variables. Variables y_d^λ denote whether demand d is accepted using wavelength λ, variables x_{de}^λ state whether edge e is used to transport demand d on wavelength λ.

$$\min \max_{e \in E} \sum_{d \in D, \lambda \in \Lambda} x_{de}^\lambda \tag{1}$$

s.t.

$$y_d^\lambda \in \{0,1\}, x_{de}^\lambda \in \{0,1\} \tag{2}$$

$$\forall d \in D: \quad \sum_{\lambda \in \Lambda} y_d^\lambda = 1 \tag{3}$$

$$\forall e \in E, \forall \lambda \in \Lambda: \quad \sum_{d \in D} x_{de}^\lambda \leq 1 \tag{4}$$

$$\forall d \in D, \forall \lambda \in \Lambda: \quad \sum_{e \in \text{In}(s(d))} x_{de}^\lambda = 0, \quad \sum_{e \in \text{Out}(s(d))} x_{de}^\lambda = y_d^\lambda \tag{5}$$

$$\forall d \in D, \forall \lambda \in \Lambda: \quad \sum_{e \in \text{Out}(t(d))} x_{de}^\lambda = 0, \quad \sum_{e \in \text{In}(t(d))} x_{de}^\lambda = y_d^\lambda \tag{6}$$

$$\forall d \in D, \forall \lambda \in \Lambda, \forall n \in N \setminus \{s(d), t(d)\}: \quad \sum_{e \in \text{In}(n)} x_{de}^\lambda = \sum_{e \in \text{Out}(n)} x_{de}^\lambda \tag{7}$$

Constraint (2) enforces integrality of the solution, constraint (3) states that all demands must be accepted and must use exactly one wavelength. The *clash* constraint (4) states that on each edge, only one demand may use any given wavelength. We further have constraints (5) and (6), which link the x and y variables at the source (resp. sink) of each demand. Finally, constraint (7) enforces flow balance on all other nodes of the network.

1.2 Extended Problem

Note that this model minimizes the maximal number of frequencies used on any link, not the overall number of frequencies. For this we have to introduce another set of 0/1 indicator variables z^λ which state whether wavelength λ is used by any demand in the network. The objective is then to minimize the sum of the z^λ variables. We also impose inequality constraints between x^λ_{de} and z^λ variables in constraint (11) of the following, *extended model* which force the indicator variable for a frequency to be set as soon as one demand uses the frequency.

It is not clear a priori whether the basic or the extended model capture the objective of minimizing the number of frequencies used, we will have to consider both alternatives in our solution approach. Both variants occur in the literature [5], without a clear indication which would be more relevant in practice.

$$\min \sum_{\lambda \in \Lambda} z^\lambda \tag{8}$$

s.t.

$$z^\lambda \in \{0,1\}, y^\lambda_d \in \{0,1\}, x^\lambda_{de} \in \{0,1\} \tag{9}$$

$$\forall d \in D: \quad \sum_{\lambda \in \Lambda} y^\lambda_d = 1 \tag{10}$$

$$\forall d \in D, \forall e \in E, \forall \lambda \in \Lambda: \quad x^\lambda_{de} \le z^\lambda \tag{11}$$

$$\forall e \in E, \forall \lambda \in \Lambda: \quad \sum_{d \in D} x^\lambda_{de} \le 1 \tag{12}$$

$$\forall d \in D, \forall \lambda \in \Lambda: \quad \sum_{e \in \text{In}(s(d))} x^\lambda_{de} = 0, \quad \sum_{e \in \text{Out}(s(d))} x^\lambda_{de} = y^\lambda_d \tag{13}$$

$$\forall d \in D, \forall \lambda \in \Lambda: \quad \sum_{e \in \text{Out}(t(d))} x^\lambda_{de} = 0, \quad \sum_{e \in \text{In}(t(d))} x^\lambda_{de} = y^\lambda_d \tag{14}$$

$$\forall d \in D, \forall \lambda \in \Lambda, \forall n \in N \setminus \{s(d), t(d)\}: \quad \sum_{e \in \text{In}(n)} x^\lambda_{de} = \sum_{e \in \text{Out}(n)} x^\lambda_{de} \tag{15}$$

Figure 2 shows the difference between the basic and extended cost on a small example with three nodes 1, 2, 3 and three demands A, B, C. On each directed link we need only two colours, that means that the basic model has cost 2, but overall we need three colours for a feasible solution, the extended model therefore has cost 3.

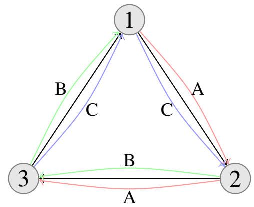

Fig. 2. Difference Between Basic and Extended Cost

1.3 Contribution and Related Work

As the complete model is quite hard to solve, it has been suggested before [1] that a two-step decomposition into a routing and a wavelength assignment phase would be a good solution technique for this problem. We re-use this idea, but strengthen it by improving each phase with some new techniques.

The main contributions of this paper are

- a comparison of different, generic solution methods for the generated graph coloring problem, using MIP, SAT and finite domain constraint programming,
- a new, very accurate lower bound to the RWA problem based on a resource-based relaxation of an existing, source aggregation MIP solution,
- experimental results showing that using constraint programming very high quality solutions are obtained by this method in seconds, significantly outperforming the other techniques,
- results indicate that the basic problem is relatively easy to solve with a variety of techniques, while the extended problem is much harder.

The RWA problem has been studied using many different solution methods, see [4] for an overview. We can distinguish two main approaches. Greedy heuristics use local search techniques to accept demands incrementally, providing fast solutions for large problem cases, but without a formal guarantee of solution quality. Alternatively, complete methods, mainly based on ILP (Integer Linear Programming) techniques, can provide optimal solutions, but are restricted in the problem size handled [5,6].

The static design problem considered here requires a somewhat different solution approach than the demand acceptance problem discussed in [13]. It uses a similar two-phase decomposition, but the relaxation of the second phase is much simpler, handled by adding additional frequencies rather than using explanation techniques to identify demands to be removed from the problem. At the same time, the resource MIP problems seems more difficult to solve for the static design case, restricting scalability with regards to network size.

A general overview of constraint applications in the network domain is given in [12]. Smith in [14] discusses a design problem for optical networks, but this is restricted to a ring topology, and minimizes the need for ADM multiplexers.

The RWA problem considered here is not too far removed from the static design problem in MPLS traffic engineering (MPLS-TE) in IP networks, which has been approached with multiple hybrid constraint solution techniques as described in [8,7,12].

The main difference is that demands in the MPLS-TE problem have integer sizes and overall link capacity limits are enforced instead of clash constraints. Note that the choice of objective function (static design vs. demand acceptance) also plays a major role in influencing the solution methods for MPLS-TE.

2 Source Aggregation

The direct formulation of the problem based on (1) or (8) does not scale well for increasing network size or number of demands. A possible improvement has been described in the literature for the RWA by aggregating flows for all demands originating in the same source node. This removes some of the symmetries that have to be considered and reduces the problem sensitivity to increasing number of demands. We can adjust the source aggregation model used in [13] based on [5] to the different objective functions discussed here, this leads to the following model for the basic problem:

$$\min z_{\max} \tag{16}$$

s.t.

$$z_{\max} \in \{0, 1...|\Lambda|\}, x^{\lambda}_{se} \in \{0,1\} \tag{17}$$

$$\forall e \in E, \forall \lambda \in \Lambda : \quad \sum_{s \in N} x^{\lambda}_{se} \leq 1 \tag{18}$$

$$\forall s \in N, \forall \lambda \in \Lambda : \quad \sum_{e \in \text{In}(s)} x^{\lambda}_{se} = 0 \tag{19}$$

$$\forall s \in N, \forall d \in D_s, \forall \lambda \in \Lambda : \quad \sum_{e \in \text{In}(d)} x^{\lambda}_{se} \geq \sum_{e \in \text{Out}(d)} x^{\lambda}_{se} \tag{20}$$

$$\forall s \in N, \forall d \in D_s : \quad \sum_{\lambda \in \Lambda} \sum_{e \in \text{In}(d)} x^{\lambda}_{se} = \sum_{\lambda \in \Lambda} \sum_{e \in \text{Out}(d)} x^{\lambda}_{se} + P_{sd} \tag{21}$$

$$\forall s \in N, \forall n \neq s, n \notin D_s, \forall \lambda \in \Lambda : \quad \sum_{e \in \text{In}(n)} x^{\lambda}_{se} = \sum_{e \in \text{Out}(n)} x^{\lambda}_{se} \tag{22}$$

$$\forall_{e \in E} : \quad \sum_{s \in N} \sum_{\lambda \in \Lambda} x^{\lambda}_{se} \leq z_{\max} \tag{23}$$

Constraints (17) define the integrality conditions. Constraint (18) specifies the *clash* constraint between demands from different sources. Constraint (19) states that demands originating in s can not be routed through s, while constraints (20) and (21) consider the destinations of demands originating in s and state that the correct number P_{sd} of demands must be dropped in each node. Constraint (22) enforces flow balance at all other nodes of the network. The integer objective value $z_{\max}$ is linked to the decision variables via the inequalities (23) which bound the cost by the maximal number of frequencies used on any link of the network.

If we change the objective function to handle the extended problem, we find that the model can no longer solve realistic problem instances.

3 Solution Approach

In this section we describe our solution approach which is based on a simple decomposition strategy already proposed in [1]. A solution to the static design RWA problem must consider the following three activities:

1. Choose path for each demand
2. Assign wavelength for each demand
3. Minimize number of wavelengths used (basic or extended model)

We choose a decomposition technique which handles the first step with a MIP program which assigns paths to the demands while minimizing the maximal number of demands routed over a link. The second and third step are expressed as a graph coloring problem where the nodes are demands and disequality constraints (edges) are imposed between any two demands which are routed over the same link in the network. The overall solution approach is shown in Figure 3. When using the MIP-MIP decomposition, the graph coloring problem is solved as an optimization problem, minimizing the number of wavelengths used. In the MIP-SAT/FD decomposition, we use a feasibility check for the graph coloring problem. We start with the minimal number of wavelengths required by the solution of the first phase. If we find a solution, the overall problem is solved to optimality. If the problem is infeasible (or the solver times out), we increase the number of wavelengths considered until we find a good, but possibly sub-optimal solution.

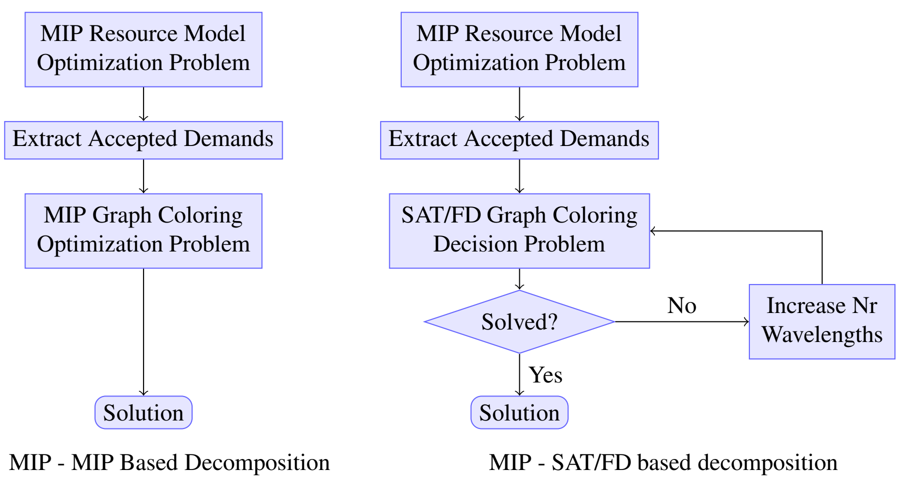

Fig. 3. Solution Approach

3.1 Phase 1

The input for phase 1 is a demand matrix, an example for the `nsf` network is shown in Figure 4. The colours encode the minimal distance between the nodes.

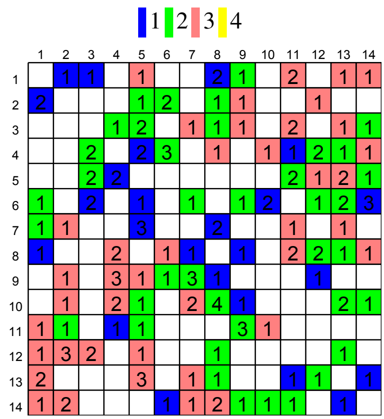

Fig. 4. Sample Demand Matrix (100 Demands) for `nsf` Network

The first phase of the decomposition is a MIP model which minimizes the maximum number of demands routed over any link in the network. The model is a relaxation of the complete model (16), obtained by ignoring allocated frequencies and instead only counting the number of demands routed over each link. Integer variables z_{se} state how many demands originating in s are routed over edge e. The domain of these variables is limited by T_s, the total number of demands originating in s.

$$\min z_{\max} \tag{24}$$

s.t.

$$z_{\max} \in \{0, 1...|\Lambda|\}, z_{se} \in \{0, 1...T_s\} \tag{25}$$

$$\forall s \in N: \quad \sum_{e \in \text{In}(s)} z_{se} = 0 \tag{26}$$

$$\forall s \in N, \forall d \in D_s: \quad \sum_{e \in \text{In}(d)} z_{se} = \sum_{e \in \text{Out}(d)} z_{se} + P_{sd} \tag{27}$$

$$\forall s \in N, \forall n \neq s, n \notin D_s: \quad \sum_{e \in \text{In}(n)} z_{se} = \sum_{e \in \text{Out}(n)} z_{se} \tag{28}$$

$$\forall_{e \in E}: \quad \sum_{s \in N} z_{se} \leq z_{\max} \tag{29}$$

Constraint (25) describes the integrality constraints, note that the variables have integer (not 0/1) domains. The clash constraint (4) has disappeared, the capacity limit for each link is handled as part of the objective function. Constraint (26) limits the use of the source node, while constraint (27) describes the balance around the destination nodes, using P_{sd}, the (fixed) number of demands from s to d. Finally, constraint (28) imposes flow balance for all other nodes. Constraints (29) link the objective to the decision variables.

The solution to (24) does not immediately return the routing for each demand, this requires a non-deterministic, but backtrack-free program to construct the paths, while at the same time removing possible loops from the solution. Figure 5 shows the result of phase 1 for source node 3, i.e. the third row in the demand matrix of Figure 4. Numbers in the nodes state how many demands originating in S end in that node, numbers on the edges state how many demands from the source are routed over them. The figure highlights a situation where we have multiple paths between the source S in node 3 and one of the destinations (node 11, marked A). We can freely choose which demand to send over which path, as long as we satisfy the capacity restrictions.

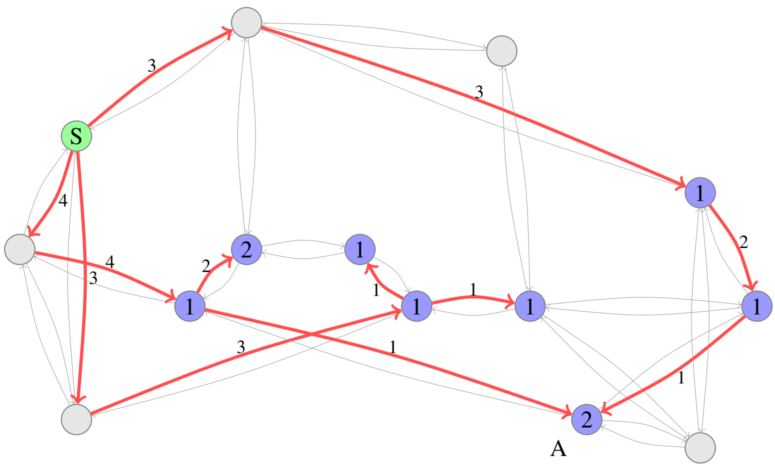

Fig. 5. Phase 1: Example Solution for Source Node 3 (Marked S)

3.2 Phase 2

The graph coloring problem for the second phase is expressed with three different solvers, a MIP optimization problem, and a SAT or finite domain decision procedure. All work on the same graph coloring instance, where each demand is a node, and two nodes (demands) are linked if they are routed over the same edge in the network. The MIP and SAT models use 0/1 integer variables x_d^λ, which state whether demand d is using wavelength λ. The finite domain model uses variables y_d which range over values 1 to Λ, the number of wavelengths considered in the model. As constraints it uses `Alldifferent` constraints instead of binary disequalities, which allows us to use stronger propagation methods [15].

Figure 6 shows the resource requirements computed for the sample demand matrix. The numbers (and colours) on the edges denote how many demands are routed over them, this corresponds to the size of the `Alldifferent` constraints required for that link. Note that the largest number of demands (13) is used on only 5 of the links. These will be the most difficult constraints to satisfy, as the number of variables is equal to the number of colours.

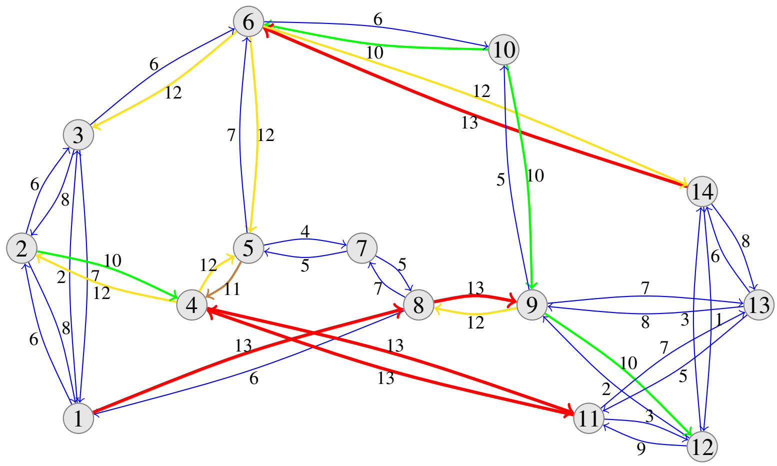

Fig. 6. Phase 2: Resource Requirements

We use the predicate $p(d, e)$ to denote if demand d was routed over edge e in the solution of the phase 1 problem.

Phase 2 MIP Formulations. The graph coloring problem leads to a simple MIP formulation for the second phase of the basic problem:

$$\min z_{\text{max}} \tag{30}$$

s.t.

$$x_d^\lambda \in \{0, 1\}, z_{\text{max}} \in \{0, 1, ..., |\Lambda|\} \tag{31}$$

$$\forall_{d \in D} : \quad \sum_{\lambda \in \Lambda} x_d^\lambda = 1 \tag{32}$$

$$\forall_{e \in E} \forall_{\lambda \in \Lambda} : \quad \sum_{\{d \in D \mid p(d,e)\}} x_d^\lambda \leq 1 \tag{33}$$

$$\forall_{e \in E} : \quad \sum_{\lambda \in \Lambda} \sum_{\{d \in D \mid p(d,e)\}} x_d^\lambda \leq z_{\text{max}} \tag{34}$$

The objective is to minimize the integer variable z_{max} which is bounded by the maximal number of frequencies used on any edge of the network. Constraint (32) states that each demand must be assigned to a frequency, constraint (33) imposes the clash condition that only one demand can use a given frequency on each link, and constraint (34) links the cost and the decision variables.

We can obtain a model for the extended problem by adding 0/1 decision variables z^λ which indicate if frequency λ is used by any of the demands. We link the new z^λ variables to the x_d^λ variables by inequality (40), and update the objective function with inequality (41). We keep inequalities (39), without them the linear relaxation only produces a very weak lower bound of 1.

$$\min z_{\max} \tag{35}$$

s.t.

$$x_d^\lambda \in \{0,1\}, z^\lambda \in \{0,1\}, z_{\max} \in \{0,1,...,|\Lambda|\} \tag{36}$$

$$\forall_{d \in D}: \quad \sum_{\lambda \in \Lambda} x_d^\lambda = 1 \tag{37}$$

$$\forall_{e \in E} \forall_{\lambda \in \Lambda}: \quad \sum_{\{d \in D \mid p(d,e)\}} x_d^\lambda \leq 1 \tag{38}$$

$$\forall_{e \in E}: \quad \sum_{\lambda \in \Lambda} \sum_{\{d \in D \mid p(d,e)\}} x_d^\lambda \leq z_{\max} \tag{39}$$

$$\forall_{d \in D} \forall_{\lambda \in \Lambda}: \quad x_d^\lambda \leq z^\lambda \tag{40}$$

$$\sum_{\lambda \in \Lambda} z^\lambda \leq z_{\max} \tag{41}$$

Phase 2 Finite Domain Model. For the finite domain model, we use variables y_d which range over all possible frequencies. To express the objective of the basic problem, we need to consider how many different frequencies are used on each link. We can use the `NValue` constraint [2] to count the number of different values used, leading to a model:

$$\min \max_{e \in E} n_e \tag{42}$$

s.t.

$$y_d \in \{0,1...,|\Lambda|\}, n_e \in \{0,1...,|\Lambda|\} \tag{43}$$

$$\forall_{e \in E}: \quad \text{nvalue}(n_e, \{y_d \mid p(d,e)\}) \tag{44}$$

$$\forall_{e \in E}: \quad \text{alldifferent}(\{y_d \mid p(d,e)\}) \tag{45}$$

Since the `NValue` and `Alldifferent` constraints are expressed over the same variable sets, the problem can be drastically simplified. We know that the values in the `Alldifferent` constraint must be pairwise different, and therefore find that the number of different values is equal to the number of variables in the constraint. The largest `Alldifferent` constraint will be set up on some link where the optimal cost was reached in phase1. The finite domain model for the basic problem therefore is no longer an optimization problem, but a feasibility problem over arbitrary domains:

$$y_d \in \{0,1...,|\Lambda|\} \tag{46}$$

$$\forall_{e \in E}: \quad \text{alldifferent}(\{y_d \mid p(d,e)\}) \tag{47}$$

For the extended problem, the phase 2 finite domain model is

$$\min \max_{d \in D} y_d \tag{48}$$

s.t.

$$y_d \in \{0,1...,|\Lambda|\} \tag{49}$$

$$\forall_{e \in E}: \quad \text{alldifferent}(\{y_d \mid p(d,e)\}) \tag{50}$$

We use "optimization from below", and try out increasing values C for the objective until we find a feasible solution. The (fixed) objective serves as upper bound on the domain of the y_d variables for each of the instances tested:

$$y_d \in \{0, 1..., C\} \quad (51)$$

$$\forall_{e \in E} : \quad \text{alldifferent}(\{y_d \mid p(d, e)\}) \quad (52)$$

Phase 2 SAT Formulation. A SAT model for the second phase can be derived using the x_d^λ variables and the clauses

$$\forall_{d \in D} \forall_{\lambda_1, \lambda_2 \in \Lambda \text{ s.t. } \lambda_1 \neq \lambda_2} : \quad \neg x_d^{\lambda_1} \vee \neg x_d^{\lambda_2} \quad (53)$$

$$\forall_{d \in D} : \quad \bigvee_{\lambda \in \Lambda} x_d^\lambda \quad (54)$$

$$\forall_{e \in E} \forall_{\lambda \in \Lambda}, d_1, d_2 \in D \text{ s.t. } p(d_1, e) \wedge p(d_2, e) \wedge d_1 \neq d_2 : \quad \neg x_{d_1}^\lambda \vee \neg x_{d_2}^\lambda \quad (55)$$

Constraints (53) state that a demand can not be assigned to more than one frequency, constraints (54) impose the other condition that each demand must be allocated to at least one frequency, and constraints (55) impose the clash constraints between any two demands routed over the same edge of the network. Alternatively, instead of the clausal representation, it is also possible to use the linear constraints of the MIP model directly in a Pseudo-Boolean solver.

4 Experimental Results

Most of the published results on the RWA problem use randomly generated demands on a few given network structures. We also use this approach and generate given numbers of demands between randomly chosen source and sink nodes. Multiple demands between the same nodes are allowed, but source and sink must be different.

In the literature we found four actual optical network topologies used in experiments. Their size is quite small, ranging from 14 to 27 nodes.

nsf 14 nodes, 42 edges
eon 20 nodes, 78 edges
mci 19 nodes, 64 edges
brezil 27 nodes, 140 edges

We explored different combinations of number of demands (100-600 demands in increments of 100) for each of the network topologies and created 100 random problem instances for each combination.

4.1 Basic Problem

The basic problem seems to be relatively well behaved, Table 1 shows some results for the complete (non-decomposed), source aggregation MIP model described in section 2.

Table 1. Selected Full MIP Examples (Basic Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg LP Gap	Max LP Gap	Avg LP Time	Max LP Time	Avg MIP Time	Max MIP Time
brezil	100	50	100	4.24	4.57	0.33	0.90	165.65	686.55	277.14	1139.03
brezil	200	50	15	7.62	7.93	0.32	0.75	585.18	2022.48	861.74	2301.67
eon	100	50	100	6.36	6.65	0.29	0.75	13.69	43.94	33.62	70.92
eon	200	50	100	11.54	11.77	0.23	0.75	27.17	147.25	65.51	257.97
eon	300	50	100	16.62	16.89	0.27	0.75	33.08	143.49	121.27	517.50
eon	400	50	100	21.47	21.85	0.38	0.75	19.87	92.49	116.64	363.53
eon	500	50	100	26.43	26.62	0.19	0.75	23.44	99.09	162.55	468.56
eon	600	50	100	31.36	31.63	0.27	0.75	28.94	73.19	232.91	542.83
mci	100	50	100	7.67	7.81	0.14	0.83	8.45	26.36	20.27	42.42
mci	200	50	100	13.42	13.58	0.16	0.80	13.45	45.88	38.79	161.02
mci	300	50	100	19.24	19.37	0.13	0.80	15.56	97.48	55.78	239.11
mci	400	50	100	25.00	25.14	0.14	0.80	18.37	58.34	109.85	484.69
mci	500	50	100	30.45	30.58	0.13	0.80	16.46	50.08	129.90	454.33
mci	600	50	100	36.00	36.11	0.11	0.80	27.50	99.06	257.70	599.44
nsf	100	50	100	7.97	8.38	0.41	0.90	3.09	5.22	8.17	14.55
nsf	200	50	100	15.06	15.45	0.39	0.75	3.36	5.44	12.75	29.45
nsf	300	50	100	21.96	22.29	0.33	0.75	3.20	5.45	17.01	39.17
nsf	400	50	100	28.81	29.18	0.37	0.75	3.49	6.31	27.36	78.66
nsf	500	50	100	35.79	36.13	0.34	0.79	4.97	11.75	54.60	125.30
nsf	600	50	100	42.52	42.94	0.42	0.75	8.06	15.84	88.72	272.26

Table 2. Selected MIP-MIP Decomposition Examples (Basic Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg MIP2	Max LP Gap	Max MIP2 Gap	Avg MIP Time	Max MIP Time	Avg MIP2 Time	Max MIP2 Time
brezil	100	150	100	4.24	4.57	4.57	0.90	0.00	0.41	0.59	0.91	1.53
brezil	200	150	100	7.92	8.26	8.26	0.75	0.00	0.46	0.58	4.45	5.97
brezil	300	150	100	11.51	11.92	11.92	0.80	0.00	0.47	0.63	8.08	9.64
brezil	400	150	100	15.10	15.45	15.45	0.75	0.00	0.51	0.70	10.93	15.84
brezil	500	150	100	18.76	19.10	19.10	0.75	0.00	0.48	0.64	13.09	17.84
brezil	600	150	100	22.32	22.61	22.61	0.75	0.00	0.51	0.66	16.77	20.56
eon	100	150	100	6.36	6.65	6.65	0.75	0.00	0.13	0.16	1.51	3.03
eon	200	150	100	11.54	11.77	11.77	0.75	0.00	0.14	0.19	5.27	7.66
eon	300	150	100	16.62	16.89	16.89	0.75	0.00	0.14	0.19	5.60	8.56
eon	400	150	100	21.47	21.85	21.85	0.75	0.00	0.17	0.22	7.38	12.11
eon	500	150	100	26.43	26.62	26.62	0.75	0.00	0.15	0.17	9.58	17.89
eon	600	150	99	31.36	31.63	31.63	0.75	0.00	0.17	0.20	14.04	27.50
mci	100	150	100	7.67	7.81	7.81	0.83	0.00	0.08	0.26	2.08	3.13
mci	200	150	100	13.42	13.58	13.58	0.80	0.00	0.09	0.09	5.36	7.69
mci	300	150	100	19.24	19.37	19.37	0.80	0.00	0.09	0.11	5.83	7.73
mci	400	150	100	25.00	25.14	25.14	0.80	0.00	0.10	0.13	8.71	12.76
mci	500	150	100	30.45	30.58	30.58	0.80	0.00	0.10	0.13	13.89	22.41
mci	600	150	100	36.00	36.11	36.11	0.80	0.00	0.11	0.14	22.56	43.58
nsf	100	150	100	7.97	8.38	8.38	0.90	0.00	0.04	0.05	2.38	3.64
nsf	200	150	100	15.06	15.45	15.45	0.75	0.00	0.04	0.06	1.81	4.39
nsf	300	150	100	21.96	22.29	22.29	0.75	0.00	0.04	0.06	1.98	6.33
nsf	400	150	100	28.81	29.18	29.18	0.75	0.00	0.06	0.08	3.54	12.34
nsf	500	150	100	35.79	36.13	36.13	0.79	0.00	0.05	0.06	5.77	9.38
nsf	600	150	100	42.52	42.94	42.94	0.75	0.00	0.06	0.08	9.09	16.19

Multiple problem instances for the brezil network with 200 or more demands did not find feasible solutions within 1 hour, for the other example networks solutions were found within 10 minutes.

The decomposition seems to work very well for the basic problem. Table 2 shows results for the MIP-MIP decomposition, Table 3 shows results for the MIP-FD decomposition, which finds the optimal solution for nearly all instances in less than a second. The SAT model (results shown in Table 4) is nearly as efficient.

Table 3. Selected Finite Domain Examples (Basic Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg FD	Max LP Gap	Max FD Gap	Avg MIP Time	Max MIP Time	Avg FD Time	Max FD Time
brezil	100	150	100	4.24	4.57	4.57	0.90	0.00	0.43	0.59	0.01	0.02
brezil	200	150	100	7.92	8.26	8.26	0.75	0.00	0.47	0.58	0.03	0.05
brezil	300	150	99	11.51	11.92	11.93	0.80	1.00	0.49	0.63	0.07	0.09
brezil	400	150	100	15.10	15.45	15.45	0.75	0.00	0.48	0.69	0.13	0.16
brezil	500	150	100	18.76	19.10	19.10	0.75	0.00	0.48	0.64	0.23	0.27
brezil	600	150	100	22.32	22.61	22.61	0.75	0.00	0.49	0.64	0.31	0.36
eon	100	150	100	6.36	6.65	6.65	0.75	0.00	0.15	0.17	0.01	0.02
eon	200	150	100	11.54	11.77	11.77	0.75	0.00	0.15	0.19	0.04	0.06
eon	300	150	100	16.62	16.89	16.89	0.75	0.00	0.16	0.19	0.09	0.11
eon	400	150	100	21.47	21.85	21.85	0.75	0.00	0.16	0.17	0.16	0.20
eon	500	150	100	26.43	26.62	26.62	0.75	0.00	0.16	0.17	0.29	0.33
eon	600	150	100	31.36	31.63	31.63	0.75	0.00	0.16	0.19	0.40	0.47
mci	100	150	100	7.67	7.81	7.81	0.83	0.00	0.10	0.19	0.01	0.02
mci	200	150	100	13.42	13.58	13.58	0.80	0.00	0.10	0.13	0.05	0.06
mci	300	150	100	19.24	19.37	19.37	0.80	0.00	0.10	0.13	0.10	0.13
mci	400	150	100	25.00	25.14	25.14	0.80	0.00	0.11	0.13	0.19	0.20
mci	500	150	100	30.45	30.58	30.58	0.80	0.00	0.11	0.13	0.29	0.41
mci	600	150	100	36.00	36.11	36.11	0.80	0.00	0.11	0.13	0.45	0.55
nsf	100	150	100	7.97	8.38	8.38	0.90	0.00	0.05	0.06	0.02	0.02
nsf	200	150	100	15.06	15.45	15.45	0.75	0.00	0.06	0.06	0.05	0.06
nsf	300	150	100	21.96	22.29	22.29	0.75	0.00	0.06	0.06	0.10	0.13
nsf	400	150	100	28.81	29.18	29.18	0.75	0.00	0.06	0.08	0.17	0.20
nsf	500	150	100	35.79	36.13	36.13	0.79	0.00	0.06	0.06	0.31	0.34
nsf	600	150	100	42.52	42.94	42.94	0.75	0.00	0.06	0.06	0.43	0.48

Table 4. Selected SAT Examples (Basic Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg SAT	Max LP Gap	Max SAT Gap	Avg MIP Time	Max MIP Time	Avg SAT Time	Max SAT Time
brezil	100	150	100	4.24	4.57	4.57	0.90	0.00	0.37	0.59	0.03	0.05
brezil	200	150	100	7.92	8.26	8.26	0.75	0.00	0.41	0.59	0.07	0.09
brezil	300	150	100	11.51	11.92	11.92	0.80	0.00	0.41	0.59	0.15	0.20
brezil	400	150	100	15.10	15.45	15.45	0.75	0.00	0.43	0.56	0.27	0.38
brezil	500	150	100	18.76	19.10	19.10	0.75	0.00	0.42	0.52	0.44	0.58
brezil	600	150	100	22.32	22.61	22.61	0.75	0.00	0.42	0.58	0.69	0.88
eon	100	150	100	6.36	6.65	6.65	0.75	0.00	0.14	0.16	0.04	0.06
eon	200	150	100	11.54	11.77	11.77	0.75	0.00	0.14	0.17	0.10	0.13
eon	300	150	100	16.62	16.89	16.89	0.75	0.00	0.14	0.17	0.24	0.31
eon	400	150	100	21.47	21.85	21.85	0.75	0.00	0.13	0.16	0.45	0.61
eon	500	150	100	26.43	26.62	26.62	0.75	0.00	0.13	0.25	0.76	1.08
eon	600	150	100	31.36	31.63	31.63	0.75	0.00	0.14	0.31	1.20	1.73
mci	100	150	100	7.67	7.81	7.81	0.83	0.00	0.13	0.23	0.05	0.08
mci	200	150	100	13.42	13.58	13.58	0.80	0.00	0.10	0.13	0.12	0.17
mci	300	150	100	19.24	19.37	19.37	0.80	0.00	0.09	0.13	0.29	0.42
mci	400	150	100	25.00	25.14	25.14	0.80	0.00	0.10	0.13	0.56	0.78
mci	500	150	100	30.45	30.58	30.58	0.80	0.00	0.10	0.27	0.97	1.41
mci	600	150	100	36.00	36.11	36.11	0.80	0.00	0.10	0.25	1.55	2.33
nsf	100	150	100	7.97	8.38	8.38	0.90	0.00	0.06	0.11	0.05	0.08
nsf	200	150	100	15.06	15.45	15.45	0.75	0.00	0.05	0.06	0.15	0.19
nsf	300	150	100	21.96	22.29	22.29	0.75	0.00	0.05	0.06	0.35	0.44
nsf	400	150	100	28.81	29.18	29.18	0.75	0.00	0.05	0.06	0.71	0.92
nsf	500	150	100	35.79	36.13	36.13	0.79	0.00	0.05	0.17	1.26	1.55
nsf	600	150	100	42.52	42.94	42.94	0.75	0.00	0.05	0.06	2.07	2.42

4.2 Extended Problem

The problem seems to be much more difficult if we consider the extended formulation minimizing the total number of frequencies used. The complete source aggregation model is not able to solve problem instances of the given sizes consistently. Table 5 shows some results for the MIP-MIP decomposition run with a timeout of 1000 seconds. After the timeout we either use the best feasible, integer solution or declare the problem as unsolved if no feasible solution has been found.

Table 5. Selected MIP-MIP Decomposition Examples (Extended Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg MIP2	Max LP Gap	Max MIP2 Gap	Avg MIP Time	Max MIP Time	Avg MIP2 Time	Max MIP2 Time
brezil	100	50	94	4.24	4.57	4.63	0.90	1.00	0.35	0.53	53.59	962.72
brezil	200	50	99	7.92	8.26	8.27	0.75	1.00	0.38	0.52	141.04	331.05
brezil	300	50	88	11.48	11.87	11.94	0.80	2.00	0.38	0.50	444.64	995.14
eon	100	50	100	6.36	6.65	6.65	0.75	0.00	0.13	0.16	19.70	61.98
eon	200	50	100	11.54	11.77	11.77	0.75	0.00	0.14	0.17	188.55	925.44
mci	100	50	100	7.67	7.81	7.81	0.83	0.00	0.09	0.11	26.27	79.55
mci	200	50	96	13.42	13.58	13.63	0.80	2.00	0.10	0.13	271.65	992.88
nsf	100	50	99	7.97	8.38	8.39	0.90	1.00	0.05	0.06	29.43	967.70
nsf	200	50	99	15.06	15.45	15.46	0.75	1.00	0.05	0.06	208.72	998.00

Table 6. Selected Finite Domain Examples (Extended Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg FD	Max LP Gap	Max FD Gap	Avg MIP Time	Max MIP Time	Avg FD Time	Max FD Time
brezil	100	150	95	4.24	4.57	4.62	0.90	1.00	0.44	0.61	0.02	0.11
brezil	200	150	99	7.92	8.26	8.27	0.75	1.00	0.48	0.59	0.06	0.09
brezil	300	150	99	11.51	11.92	11.94	0.80	2.00	0.49	0.64	0.12	0.19
brezil	400	150	99	15.10	15.45	15.46	0.75	1.00	0.50	0.69	0.23	0.31
brezil	500	150	96	18.76	19.10	19.16	0.75	3.00	0.50	0.66	0.93	60.63
brezil	600	150	97	22.32	22.61	22.64	0.75	1.00	0.51	0.64	0.45	0.97
eon	100	150	100	6.36	6.65	6.65	0.75	0.00	0.15	0.16	0.02	0.03
eon	200	150	100	11.54	11.77	11.77	0.75	0.00	0.16	0.19	0.07	0.16
eon	300	150	100	16.62	16.89	16.89	0.75	0.00	0.16	0.19	0.16	0.24
eon	400	150	100	21.47	21.85	21.85	0.75	0.00	0.16	0.19	0.26	0.38
eon	500	150	100	26.43	26.62	26.62	0.75	0.00	0.17	0.22	0.44	0.64
eon	600	150	100	31.36	31.63	31.63	0.75	0.00	0.17	0.20	0.60	0.98
mci	100	150	100	7.67	7.81	7.81	0.83	0.00	0.10	0.19	0.02	0.05
mci	200	150	100	13.42	13.58	13.58	0.80	0.00	0.10	0.13	0.08	0.13
mci	300	150	100	19.24	19.37	19.37	0.80	0.00	0.11	0.13	0.17	0.55
mci	400	150	100	25.00	25.14	25.14	0.80	0.00	0.11	0.14	0.32	0.58
mci	500	150	100	30.45	30.58	30.58	0.80	0.00	0.11	0.14	0.48	1.47
mci	600	150	100	36.00	36.11	36.11	0.80	0.00	0.12	0.14	0.68	1.20
nsf	100	150	99	7.97	8.38	8.39	0.90	1.00	0.06	0.08	0.03	0.03
nsf	200	150	100	15.06	15.45	15.45	0.75	0.00	0.06	0.06	0.07	0.13
nsf	300	150	100	21.96	22.29	22.29	0.75	0.00	0.06	0.06	0.15	0.19
nsf	400	150	100	28.81	29.18	29.18	0.75	0.00	0.06	0.08	0.26	0.55
nsf	500	150	100	35.79	36.13	36.13	0.79	0.00	0.06	0.08	0.42	0.53
nsf	600	150	100	42.52	42.94	42.94	0.75	0.00	0.06	0.08	0.58	0.74

Table 6 shows the result for the finite domain solver. The entries summarize the results over 100 runs with the same parameters, but different random seeds. The column *Opt.* tells how many solutions were proven optimal. The columns *Avg LP*, *Avg MIP* and *Avg FD* show the average cost obtained by the LP relaxation of the first phase MIP

Table 7. Selected SAT Examples (Extended Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg SAT	Max LP Gap	Max SAT Gap	Avg MIP Time	Max MIP Time	Avg SAT Time	Max SAT Time
brezil	100	150	96	4.24	4.57	4.61	0.90	1.00	0.38	0.63	0.02	0.05
brezil	200	150	99	7.92	8.26	8.27	0.75	1.00	0.41	0.56	0.06	2.39
brezil	300	150	98	11.51	11.92	11.95	0.80	2.00	0.42	0.61	3.09	200.08
brezil	400	150	99	15.10	15.45	15.46	0.75	1.00	0.42	0.58	1.21	100.22
brezil	500	150	95	18.76	19.10	19.17	0.75	3.00	0.42	0.53	7.83	300.48
brezil	600	150	82	22.32	22.61	22.79	0.75	1.00	0.42	0.56	21.69	170.00
eon	100	150	100	6.36	6.65	6.65	0.75	0.00	0.14	0.19	0.02	0.03
eon	200	150	100	11.54	11.77	11.77	0.75	0.00	0.15	0.17	0.06	0.11
eon	300	150	100	16.62	16.89	16.89	0.75	0.00	0.15	0.17	0.19	0.39
eon	400	150	100	21.47	21.85	21.85	0.75	0.00	0.15	0.17	0.57	1.58
eon	500	150	87	26.43	26.62	26.75	0.75	1.00	0.15	0.22	15.32	102.84
eon	600	150	42	31.36	31.63	32.24	0.75	2.00	0.15	0.19	66.10	202.98
mci	100	150	100	7.67	7.81	7.81	0.84	0.00	0.10	0.20	0.02	0.03
mci	200	150	100	13.42	13.58	13.58	0.80	0.00	0.10	0.13	0.08	0.14
mci	300	150	100	19.24	19.37	19.37	0.80	0.00	0.10	0.13	0.27	0.64
mci	400	150	97	25.00	25.14	25.17	0.80	1.00	0.10	0.13	4.15	100.77
mci	500	150	78	30.45	30.58	30.80	0.80	1.00	0.10	0.13	24.33	103.81
mci	600	150	33	36.00	36.11	36.87	0.80	2.00	0.11	0.20	76.84	204.42
nsf	100	150	99	7.97	8.38	8.39	0.90	1.00	0.05	0.08	0.09	6.55
nsf	200	150	100	15.06	15.45	15.45	0.75	0.00	0.05	0.06	0.10	0.22
nsf	300	150	100	21.96	22.29	22.29	0.75	0.00	0.06	0.08	0.48	1.70
nsf	400	150	90	28.81	29.18	29.28	0.75	1.00	0.06	0.06	11.46	110.38
nsf	500	150	41	35.79	36.13	36.81	0.79	2.00	0.06	0.16	70.70	218.00
nsf	600	150	23	42.52	42.94	43.93	0.75	3.00	0.06	0.09	104.04	301.59

resource model, the MIP model itself and the total number of frequencies required by the finite domain solver. The LP relaxation already is a very good approximation of the total cost, the MIP-LP gap never exceeds 0.90. The next column, *Max FD Gap*, shows the largest gap between MIP and FD solution, i.e. the number of frequencies added due to infeasibility or time out of the graph coloring model. This value never exceeds 3 in the examples shown. We then show average and maximal run times for the first and second phases of the decomposition on a Windows XP laptop with a 2.4GHz processor and 2GB of memory. Results were obtained using ECLiPSe 6.0 [16] with the eplex library [11] for the Coin-OR [9] CLP/CBC MIP solver.

Table 7 show corresponding results for the SAT model using minisat 1.14 [3], with a timeout of 100 seconds for each instance and each tested upper bound of the domain. If a timeout occurs, the problem is re-run adding frequencies until a solution is found within the timeout period. For increasing problem sizes the number of optimal solutions decreases sharply, in contrast to the finite domain model, while execution times are increasing significantly.

The hybrid model using the finite domain model is able to deal with much larger number of demands, as Table 8 shows. We consider the brezil network, and increase the number of demands up to 2000. The solving time for the first phase is not affected, as the model is not dependent on the number of demands, they only affect the upper bound of the domains T_s and the size of the coefficients P_{sd}. In the second phase the number of variables increases with the number of demands, and the `Alldifferent` constraints operate on larger number of variables, but the number of constraints is given by the topology and does not change.

Table 8. Increasing Demand Number (Extended Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg FD	Max LP Gap	Max FD Gap	Avg MIP Time	Max MIP Time	Avg FD Time	Max FD Time
brezil	700	150	97	25.69	26.06	26.13	0.75	3.00	0.51	0.64	1.83	60.59
brezil	800	150	96	29.34	29.66	29.72	0.75	3.00	0.50	0.59	1.42	60.95
brezil	900	150	98	32.81	33.14	33.17	0.75	2.00	0.50	0.61	1.30	31.36
brezil	1000	150	99	36.34	36.68	36.69	0.75	1.00	0.50	0.63	1.24	2.13
brezil	1100	150	99	39.80	40.16	40.17	0.75	1.00	0.50	0.63	1.49	2.20
brezil	1200	150	99	43.28	43.61	43.62	0.75	1.00	0.50	0.63	2.24	46.16
brezil	1300	150	98	46.54	46.89	46.94	0.75	3.00	0.50	0.61	3.03	64.45
brezil	1400	150	99	49.85	50.21	50.23	0.75	2.00	0.50	0.63	2.79	33.95
brezil	1500	150	99	53.46	53.87	53.89	0.75	2.00	0.50	0.61	3.18	34.47
brezil	1600	150	98	56.95	57.28	57.30	0.75	1.00	0.50	0.59	4.49	72.05
brezil	1700	150	99	60.33	60.65	60.66	0.75	1.00	0.51	0.64	3.61	8.92
brezil	1800	150	99	63.93	64.25	64.26	0.75	1.00	0.51	0.61	4.08	9.49
brezil	1900	150	100	67.41	67.77	67.77	0.75	0.00	0.50	0.61	4.73	10.48
brezil	2000	150	99	70.83	71.09	71.10	0.75	1.00	0.51	0.66	6.05	94.73

Table 9. Increasing Network Size (Extended Problem, 100 Runs Each)

Network	Dem.	λ	Opt.	Avg LP	Avg MIP	Avg FD	Max LP Gap	Max FD Gap	Avg MIP Time	Max MIP Time	Avg FD Time	Max FD Time
r30	500	30	100	7.81	8.12	8.12	0.97	0.00	1.73	5.92	0.16	0.27
r40	500	30	100	4.14	4.52	4.52	0.92	0.00	12.42	177.45	0.13	0.19
r50	500	30	97	2.39	2.88	2.91	0.95	1.00	77.35	696.73	0.11	0.14
r60	500	30	100	1.57	2.05	2.05	0.86	0.00	127.75	245.25	0.10	0.13

The model is much more dependent on the size of the network. We consider in Table 9 random networks with 30-60 nodes, with a 0.25 probability for a link between two nodes. The times for the MIP (first) part of the model increase quickly with network size, and soon dominate the total execution times, while the second phase is barely affected. It is interesting that the execution times increase much more rapidly for the static design problem than for the demand acceptance problem discussed in [13], where network sizes up to 100 nodes can be solved within 30 seconds using the same environment.

5 Summary

In this paper we have considered some variants of the routing and wavelength assignment problem for optical networks. For the static design problem two possible objective functions have been proposed in the literature: The basic problem minimizing the maximal number of frequencies required on any link, and the extended problem minimizing the total number of frequencies used in the network. A decomposition into a MIP based routing part with a graph coloring second phase works very well in producing high quality solutions. Both the LP relaxation and the MIP solution of the first phase produce very accurate lower bounds on the total cost. The graph coloring problem in the basic model can be solved successfully by either MIP, SAT or finite domain constraint programming, constraint programming is slightly faster than SAT on the problem instances considered, and significantly faster than MIP. For the extended problem, constraint programming is much more stable and significantly faster than any of the competing methods.

Together with the results in [13] this shows that a decomposition of the RWA problem into MIP and FD phases can be highly successful, producing proven optimal or near-optimal results for a large set of problem instances.

References

1. Banerjee, D., Mukherjee, B.: A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. IEEE Journal on Selected Areas in Communications 14(5), 903–908 (1996)
2. Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., Walsh, T.: Filtering algorithms for the nvalue constraint. Constraints 11(4), 271–293 (2006)
3. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
4. Jaumard, B., Meyer, C., Thiongane, B.: ILP formulations for the routing and wavelength assignment problem: Symmetric systems. In: Resende, M., Pardalos, P. (eds.) Handbook of Optimization in Telecommunications, pp. 637–677. Springer, Heidelberg (2006)
5. Jaumard, B., Meyer, C., Thiongane, B.: Comparison of ILP formulations for the RWA problem. Optical Switching and Networking 4(3-4), 157–172 (2007)
6. Jaumard, B., Meyer, C., Thiongane, B.: On column generation formulations for the RWA problem. Discrete Applied Mathematics 157, 1291–1308 (2009)
7. Lever, J.: A local search/constraint propagation hybrid for a network routing problem. International Journal on Artificial Intelligence Tools 14(1-2), 43–60 (2005)
8. Liatsos, V., Novello, S., El Sakkout, H.: A probe backtrack search algorithm for network routing. In: Proceedings of the Third International Workshop on Cooperative Solvers in Constraint Programming, CoSolv 2003, Kinsale, Ireland (September 2003)
9. Lougee-Heimer, R.: The common optimization interface for operations research. IBM Journal of Research and Development 47, 57–66 (2003)
10. Ramaswami, R., Sivarajan, K.N.: Routing and wavelength assignment in all-optical networks. IEEE/ACM Trans. Netw. 3(5), 489–500 (1995)
11. Shen, K., Schimpf, J.: Eplex: Harnessing mathematical programming solvers for constraint logic programming. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 622–636. Springer, Heidelberg (2005)
12. Simonis, H.: Constraint applications in networks. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
13. Simonis, H.: A hybrid constraint model for the routing and wavelength assignment problem. In: Gent, I.P. (ed.) CP 2009. LNCS, vol. 5732, pp. 104–118. Springer, Heidelberg (2009)
14. Smith, B.M.: Symmetry and search in a network design problem. In: Barták, R., Milano, M. (eds.) CPAIOR 2005. LNCS, vol. 3524, pp. 336–350. Springer, Heidelberg (2005)
15. van Hoeve, W.J.: The alldifferent constraint: A survey. CoRR, cs.PL/0105015 (2001)
16. Wallace, M., Novello, S., Schimpf, J.: ECLiPSe: A platform for constraint logic programming. ICL Systems Journal 12(1) (May 1997)
17. Zang, H., Jue, J.P., Mukherjee, B.: A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. Optical Networks Magazine, 47–60 (January 2000)