A Boolean Satisfiability based Solution to the Routing and Wavelength Assignment (RWA) Problem in Optical Telecommunication Networks

John Valavi[†], Nikhil Saluja[†], Sunil P Khatri[*]

[†] (valavi,saluja)@colorado.edu

[*] sunil@ee.tamu.edu

[†] Department of Electrical and Computer Engineering

University of Colorado

Boulder, CO 80309

[*] Department of Electrical Engineering

Texas A&M University

College Station, TX 77843

Outline

- Motivation and Introduction
- Prior RWA Approaches
- Boolean SATisfiability (SAT)
- SAT based RWA
  - Definitions and Terminology
  - Formulation
  - Results
- Conclusions, Future Work

Motivation and Introduction

- Dense Wavelength Division Multiplexing (DWDM) effectively multiplies bandwidth in an optical fiber by transmitting data along several wavelengths.
- Routing and Wavelength Assignment (RWA) is an important problem to be addressed in this context.
  - Data routed along a set of *lightpaths*
  - Lightpaths sharing a common link must use different wavelengths.
  - Given pattern of connection requests, need optimal routing and wavelength assignment so as to maximize throughput, while utilizing a minimum number of wavelengths.
- Variants of the RWA problem
  - With or without *wavelength translation*
  - *Static or Dynamic* RWA

Routing and Wavelength Assignment (RWA)

- We cast the RWA problem as a Boolean Satisfiability (SAT) instance, and use fast SAT solvers to perform the RWA.
  - Formulation is extremely flexible:
    - Can handle static or dynamic RWA
    - Can handle RWA with or without wavelength translation
    - Can handle arbitrary network topologies

3-4 orders of magnitude speedup compared to prior art

Previous Work

- Many approaches based on ILP, with large runtimes.
- Several heuristic approaches, such as
  - Tabu search based, for networks which allow wavelength translation
  - Genetic algorithm based
  - IP based, applicable for ring networks
- Hard in general to compare techniques since randomly generated data is utilized.

Our approach is applicable for arbitrary network topologies, and also handles wavelength translation and static/dynamic RWA in a common mathematical framework

Boolean SATisfiability

Definition 1 *A conjunctive normal form (CNF) Boolean formula f on n Boolean variables $x_1, x_2, ..., x_n$ is a conjunction (logical AND) of m clauses $c_1, c_2, ..., c_m$. Each clause c_i is the disjunction (logical OR) of its constituent literals.*

For example $f = (x_1 + x_3) \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3})$ is a CNF formula with two clauses, $c_1 = (x_1 + x_3)$ and $c_2 = (\overline{x_1} + \overline{x_2} + \overline{x_3})$.

Definition 2 *Boolean satisfiability (SAT) is the problem of determining whether a Boolean formula in conjunctive normal form (CNF) has a satisfying assignment.*

In the above example, a satisfying assignment of variables for the formula f is $x_1 = 1, x_2 = 0$.

Boolean SATisfiability ... 2

- Based on the problem instance, the SAT solver may return one of three conditions.
  - Problem is not satisfiable (solver mentions this)
  - Problem is satisfiable (solver returns *a* satisfying solution
  - Solver may timeout before concluding either of the above.
- SAT is the classic NP complete problem
- There are several heuristic solvers which are very efficient
  - GRASP, which introduced the idea of non-chronological backtrack
  - Zchaff, which introduces ”2-watched” literals for efficiency
  - CirCUs, Berkmin and others which still use the non-chronological backtrack idea of GRASP

Definitions and Terminology

We model an optical network N as a graph $G(V, E)$. An edge e_{ij} exists in G if a fiber exists between nodes i and j in N.
The k^{th} connection request (between nodes i and j in N), is represented as $R_k \equiv (v_i, v_j)$

Definition 3 *The Boolean variable $v_i^{R_k \lambda_p}$ represents the logical condition of whether a node i is part of the k^{th} connection request R_k using wavelength λ_p.*

Definition 4 *The Boolean variable $e_{ij}^{R_k \lambda_p}$ represents the logical condition of whether the edge connecting nodes i and j utilizes wavelength λ_p for the k^{th} connection request R_k. If $e_{ij}^{R_k \lambda_p} = 1$, we refer to the edge e_{ij} as an active edge.*

Definition 5 *The Boolean variable $v_i^{R_k}$ represents the logical condition of whether the node i is part of the connection request R_k. If $v_i^{R_k} = 1$, we refer to the node i as an active node*

SAT Based RWA - Formulation

- We write clauses to encode the constraints and requirements imposed by the RWA problem.
- Clauses written for a fixed number q of wavelengths.
- The different types of clauses are described next (for the case of RWA with wavelength translation allowed)
- In general, if we have a constraint of the type $a \Rightarrow b$, the corresponding clause for this condition is $(\overline{a} + b)$.
- The final CNF expression is the SAT instance that is to be solved.
- We use the Zchaff SAT solver. If the problem has no solution, we increment q and repeat the above process.

SAT Based RWA - Clause Generation

- The start node must have at least one active edge per route

$$\left(\bigvee_{p=1}^{q} \bigvee_{x \in adj\_nodes(i)} e_{ix}^{R_k \lambda_p} \right) \tag{1}$$

Such clauses are written for all routes R_k where v_i is the start node of the route.

- The end node must have at least one active edge per route

$$\left(\bigvee_{p=1}^{q} \bigvee_{x \in adj\_nodes(i)} e_{ix}^{R_k \lambda_p} \right) \tag{2}$$

Such clauses are written for all routes R_k where v_i is the end node of the route.

SAT Based RWA - Clause Generation ... 2

- The start node must have at most one active edge per route

$$\left(e_{ix}^{R_k\lambda_r} \Rightarrow \left(\prod_{\substack{t=1\\ t\neq r}}^{q} \overline{e_{ix}^{R_k\lambda_t}}\right) \prod_{p=1}^{q} \prod_{y \in adj\_nodes(i)} \overline{e_{iy}^{R_k\lambda_p}} \right) \quad ; y \neq x \qquad (3)$$

  Such clauses are written for all routes R_k and all wavelengths λ_p where v_i is the start node of the route.

- The end node must have at most one active edge per route

$$\left(e_{ix}^{R_k\lambda_r} \Rightarrow \left(\prod_{\substack{t=1\\ t\neq r}}^{q} \overline{e_{ix}^{R_k\lambda_t}}\right) \prod_{p=1}^{q} \prod_{y \in adj\_nodes(i)} \overline{e_{iy}^{R_k\lambda_p}} \right) \quad ; y \neq x \qquad (4)$$

  Such clauses are written for all routes R_k and all wavelengths λ_p where v_i is the end node of the route.

SAT Based RWA - Clause Generation ... 3

- If a light edge adjoining a node is active, then at least one other light edge adjoining the same node must be active (excluding start and end node)

$$\left(e_{ix}^{R_k \lambda_p} \Rightarrow \bigvee_{p=1}^{q} \bigvee_{y \in adj\_nodes(i)} e_{iy}^{R_k \lambda_p} \right) \quad ; x \neq y \tag{5}$$

  Such clauses are written for all routes R_k where v_i is neither start nor end node.

- The start node must be active

$$\left(v_i^{R_k} = 1 \right) \tag{6}$$

  Such clauses are written for all routes R_k where v_i is the start node.

SAT Based RWA - Clause Generation ... 4

- At most two edges adjoining a node can be active (excluding start and end node)

$$e_{ix}^{R_k\lambda_p} \cdot e_{iy}^{R_k\lambda_r} \Rightarrow (\prod_{\forall s \neq p} \overline{e_{ix}^{R_k\lambda_s}})(\prod_{\forall t \neq r} \overline{e_{iy}^{R_k\lambda_t}})$$

$$(\prod_{z \in adj\_nodes(i)} \prod_{u=1}^{q} \overline{e_{iz}^{R_k\lambda_u}}) \quad ; z \neq x, z \neq y \tag{7}$$

  Such clauses are written for all routes R_k where v_i is neither start nor end node.

- The end node must be active

$$\left(v_i^{R_k} = 1\right) \tag{8}$$

  Such clauses are written for all routes R_k where v_i is the end node.

SAT Based RWA - Clause Generation ... 5

- If a node is active and an edge connected to it is active, then the node at the other end of the edge must also be active

$$\prod_{x \in adj\_nodes(i)} \left(v_i^{R_k} \cdot \bigvee_{p=1}^{q} e_{ix}^{R_k \lambda_p} \Rightarrow v_x^{R_k} \right) \tag{9}$$

  Such clauses are written for all routes R_k.

- If two nodes are active, then the light edge connecting them must be active (in some wavelength)

$$\prod_{x \in adj\_nodes(i)} \left(v_i^{R_k} \cdot v_x^{R_k} \Rightarrow \bigvee_{p=1}^{q} e_{ix}^{R_k \lambda_p} \right) \tag{10}$$

SAT Based RWA - Clause Generation ... 6

- If a node is not active then all its adjoining edges are not active

$$\left(\overline{v_i^{R_k}} \Rightarrow \prod_{x \in adj\_nodes(i)} \prod_{p=1}^{q} \overline{e_{ix}^{R_k \lambda_p}}\right) \tag{11}$$

Such clauses are written for all routes R_k.

- If a light edge is chosen in one connection request, then it cannot be chosen in any other connection request

$$\prod_{p=1}^{q} \left(e_{ij}^{R_k \lambda_p} \Rightarrow \prod_{x=1, x \neq k}^{n} \overline{e_{ij}^{R_x \lambda_p}}\right) \tag{12}$$

Such clauses are written for all routes R_k.

SAT Based RWA - Clause Generation ... 7

- Similarly, we can write clauses for the RWA problem in which wavelength translation is not allowed.
- The number of Boolean variables in the problem is $O(|E| \cdot Q \cdot K)$, where Q is the number of wavelengths, and K is the number of connection requests.
- The number of clauses in the problem is $O(Q \cdot K)$

SAT Based RWA - Results

- Implemented in C++, using Zchaff SAT solver
- For a given RWA problem instance, first we create SAT clauses for this instance.
- Start with $q = 1$, increase until Zchaff returns a satisfying solution

Network	With Wavelength Translation				
	Variables	Clauses	Edges	Wavelength	Time in secs
A01	405	5317	19	3	0.001
A02	405	5332	19	3	0.001
ATT01	1734	41346	37	3	0.02
J01	1120	20273	30	4	0.01
J02	3000	73832	50	5	0.05
J03	1320	23957	39	4	0.02
EURO1	4740	145998	60	5	0.03

SAT Based RWA - Results ... 2

Network	Without Wavelength Translation				
	Variables	Clauses	Edges	Wavelength	Time in secs
A01	297	7068	15	3	0.001
A02	297	7022	14	3	0.001
ATT01	1360	96410	29	3	0.01
J01	784	32861	29	4	0.03
J02	2040	147522	47	5	0.02
J03	915	42548	33	4	0.01
EURO1	3474	573138	47	5	0.11

Conclusions and Future Work

- Formulated RWA as a SAT instance, and solved using efficient SAT solver
- Formulation is general
  - No restriction on network topology
  - Can handle RWA with or without wavelength translation
  - Can handle static or dynamic RWA
  - Can handle time-varying network topologies, link capacities or connection requests in an incremental manner, without perturbing previously computed solution (if so desired).
- Results demonstrate dramatic 3-4 orders of magnitude speedup over existing techniques.