Routing and Wavelength Assignment in Optical Networks Using Boolean Satisfiability

Fadi A. Aloul
Computer Engineering Department
American University of Sharjah
faloul@aus.edu

Bashar Al-Rawi
Dept. of Computer Science & Engineering
University of Michigan - Ann Arbor
brawi@umich.edu

Mokhtar Aboelaze
Dept. of Computer Science & Engineering
York University
aboelaze@cse.yorku.ca

Abstract — **Optical networks consist of switches that are connected using fiber optics links. Each link consists of a set of wavelengths and each wavelength can be used by one or more users to transmit information between two switches. In order to establish a connection between the source and destination nodes, a set of switches and links must be efficiently selected. This is known as the *routing* problem. A wavelength is then assigned in each selected link to establish the connection. This is known as the *wavelength assignment* problem. The problem of routing and wavelength assignment (RWA) in optical networks has been shown to be NP-Complete. In this paper, we propose a new approach to solving the RWA problem using advanced Boolean satisfiability (SAT) techniques. SAT has been heavily researched in the last few years. Significant advances have been proposed and have lead to the development of powerful SAT solvers that can handle very large problems. SAT solvers use intelligent search algorithms that can traverse the search space and efficiently prune parts that contain no solutions. These solvers have recently been used to solve many challenging problems in Engineering and Computer Science. In this paper, we show how to formulate the RWA problem as a SAT instance and evaluate several advanced SAT techniques in solving the problem. Our approach is verified on various network topologies. The results are promising and indicate that using the proposed approach can improve on previous techniques.**

I. INTRODUCTION

The Internet has started in the late 1970, ever since its inception, the traffic, the nodes, and the users are growing at an unprecedented pace. Today the Internet handles completely different types of traffic compared to the 1970's and 1980's Internet. Link bandwidth and Internet traffic are continuously increasing. Routing protocols are constantly being proposed and improved in order to handle the constant changes in the traffic, link bandwidth, and the required quality of service in today's Internet.

The increasing growth of Internet traffic has put a significant demand on the Internet capacity, especially in the core network [24]. *Optical* networks are gaining wide acceptance and has emerged as the solution for increasing Internet traffic demands [27], especially as backbone network for service providers.

Optical networks have a huge bandwidth advantage, security, and ease of configuration for a point to point connection which means easier Virtual Private Networks (VPN) implementation over public switching networks. Optical networks consist of fiber optic links connected by using optical switches. Since the capacity of the fiber optic link is huge, wave-length division multiplexing (WDM) is used where the bandwidth is usually divided into several *wavelengths* (or channels) and every wavelength is assigned to one user; thus sharing the link bandwidth among many users.

Another advantage of optical networks is the limitations of the electronic switches when operating at very high data rates. The circuits used at rates more than 10Gbps are very expensive, optical switches can handle such speeds (and more) with less complexities and lower cost.

The problem of *routing* and *wavelength assignment* (RWA) in optical networks is NP-complete. Methods proposed to solve the RWA problem can be classified into two different categories. The first category divides the RWA problem into two subproblems, solves each subproblem disjointly, and then combines their solutions. The second category solves both problems jointly. This requires more complex algorithms but produces better results. In this paper, we are interested in the latter approach. We show how to formulate the RWA problem as a Boolean Satisfiability (SAT) instance and explore the possibility of using advanced SAT techniques to solve the RWA problem.

Recently, SAT have been shown to be very successful in solving complex problems in various Engineering and Computer Science applications. Such applications include: Formal Verification [5], FPGA routing [23], Power Optimization [3], etc. SAT has also been extended to a variety of applications in Artificial Intelligence including other well known NP-complete problems such as graph colorability, vertex cover, hamiltonian path, and independent sets [11]. Despite SAT being an NP-Complete problem [10], many researchers have developed powerful SAT solvers that are able of handling problems consisting of thousands of variables and millions of constraints. Briefly defined, the SAT problem consists of a set of Boolean variables and a set of constraints expressed in product-of-sum form. The goal is to identify an assignment to the variables that would satisfy all constraints or prove that no such assignment exists.

In this paper, we present a SAT-based approach to solving the RWA problem in optical networks. We show how to formulate the problem as a SAT instance. We report results using randomly-generated network topologies. Initial results indicate the efficiency of the proposed approach. The proposed approach is *complete* and is guaranteed to identify the *cheapest* path and wavelength assignment, if one exists. The approach also allows user-specific conditions to be easily added to the problem, which was not as easy to add in previous approaches.

This paper is organized as follows. Section II provides a general overview of SAT. Section III shows how to formulate routing and wavelength assignments in optical networks as a SAT instance. Experimental results for optical networks are presented and discussed in Section IV. Finally, the paper is concluded in Section V.

II. BOOLEAN SATISFIABILITY

The last few years have seen significant advances in Boolean satisfiability (SAT) solving. These advances have lead to the successful deployment of SAT solvers in a wide range of problems in Engineering and Computer Science. Given a set of Boolean variables and a set of constraints expressed in product-of-sum form, the goal is to find a variable assignment that satisfies all constraints or prove that no such assignment exists.

The SAT problem is usually expressed in conjunctive normal form (CNF). A CNF formula φ on n binary variables $x_1, \ldots, x_n$ is the conjunction (AND) of m clauses $\omega_1, \ldots, \omega_m$ each of which is a disjunction (OR) of one or more literals, where a literal is the occurrence of a variable or its complement.

A variable x is said to be *assigned* when its logical value is set to 0 or 1 and *unassigned* otherwise. A literal l is a *true* (*false*) literal if it evaluates to 1 (0) under the current assignment to its associated variable, and a *free literal* if its associated variable is *unassigned*. A clause is said to be *satisfied* if at least one of its literals is true, *unsatisfied* if all of its literals are set to false, *unit* if all but a single literal are set to false, and *unresolved* otherwise. A formula is said to be satisfied if all its clauses are satisfied, and unsatisfied if at least one of its clauses is unsatisfied. In general, the SAT problem is defined as follows: Given a Boolean formula in CNF, find an assignment of variables that satisfies the formula or prove that no such assignment exists.

In the following example, the CNF formula:

$$\varphi = (a \vee b)(\bar{b} \vee c)(\bar{a} \vee c) \quad (1)$$

consists of 3 variables, 3 clauses, and 6 literals. The assignment $\{a = 1, b = 0, c = 0\}$ violates the third clause and unsatisfies φ, whereas the assignment $\{a = 1, b = 0, c = 1\}$ satisfies φ. Note that a problem with n variables will have 2^n possible assignments to test. The above example with 3 variables has 8 possible assignments.

Despite the SAT problem being NP-Complete [10], there have been dramatic improvements in SAT solver technology over the past decade. This has lead to the development of several powerful SAT algorithms that are capable of solving problems consisting of thousands of variables and millions of constraints. Such solvers include: GRASP [20], zChaff [22], and Berkmin [17]. In the next three sections, we describe the basic SAT search algorithm, recent extensions to the SAT solver input, and the use of hardware with SAT.

A. Backtrack Search

Most modern complete SAT algorithms can be classified as enhancements to the basic Davis-Logemann-Loveland (DLL) backtrack search approach [13]. The DLL procedure performs a search process that traverses the space of 2^n variable assignments until a satisfying assignment is found (the formula is satisfiable), or all combinations have been exhausted (the formula is unsatisfiable). It maintains a *decision tree* to keep track of variable assignments and can be viewed as consisting of three main engines: (1) *Decision* engine that makes *elective* assignments to the variables, (2) *Deduction* engine that determines the consequences of these assignments, typically yielding additional *forced* assignments to, i.e. implications of, other variables, and (3) *Diagnosis* engine that handles the occurrence of conflicts, i.e. assignments that cause the formula to become unsatisfiable, and backtracks appropriately.

Recent studies have proposed the use of the *conflict analysis* procedure in the diagnosis engine [20]. The idea is whenever a conflict is detected, the procedure analyzes the variable assignments that cause one or more clauses to become unsatisfied. Such analysis can identify a small subset of variables whose current assignments can be blamed for the conflict. These assignments are turned into a *conflict-induced clause* and augmented with the clause database to avoid regenerating the same conflict in future parts of the search process. In essence, the procedure performs a form of learning from the encountered conflicts. Today, conflict analysis is implemented in almost all SAT solvers [17, 20, 22].

B. More Expressive Input

Restricting the input of SAT solvers to CNF formulas can restrict their usage in various domains. Therefore, researchers have focused on extending SAT solvers to handle stronger input representations. Specifically, SAT solvers [2, 7, 14, 15, 29] have recently been extended to handle pseudo-Boolean (PB) constraints which are linear inequalities with integer coefficients that can be expressed in the normalized form [2] of:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq b \quad (2)$$

where $a_i, b \in Z^+$ and x_i are literals of Boolean variables. Note that any CNF clause can be viewed as a PB constraint, e.g. clause $(a \vee b \vee c)$ is equivalent to $(a + b + c \geq 1)$.

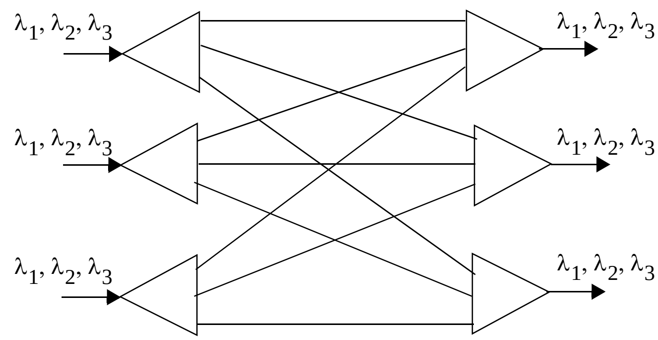

Fig. 1. An example of a wavelength router with three input and three output links. Each link has 3 wavelengths.

PB constraints can, in some cases, replace an exponential number of CNF constraints. They have been found to be very efficient in expressing "counting constraints" [2]. Furthermore, PB extends SAT solvers to handle *optimization* problems as opposed to only *decision* problems. Subject to a given set of CNF and PB constraints, one can request the minimization (or maximization) of an objective function which consists of a linear combination of the problem's variables.

$$\sum_{i=1}^{n} a_ix_i \quad (3)$$

This feature has introduced many new applications to the SAT domain. Recent studies have also shown that SAT-based optimization solvers can in fact compete with the best generic integer linear programming (ILP) solvers [2, 7].

C. Hardware-Based SAT Solvers

Note that SAT solvers can be implemented in hardware. Several studies proposed the use of FPGA reconfigurable systems to solve SAT problems [1, 33]. Hardware solvers could be a standalone or as an accelerator where the problem is partitioned between the hardware solver and the attached computer using software. Many different architecture were proposed to solve SAT problems in hardware. Linearly connected set of finite state machines, control unit, and deduction logic was proposed in [33]. The authors in [33] implemented their algorithm on Xilinx XC4028 FPGA. While in [1], the authors proposed a technique for modeling any boolean expression. Their objective is to set the function output to 1. A backtrack algorithm is used to propagate the output back to the input and finding an assignment of the inputs to satisfy a logical 1 at the output.

The authors in [12] proposed an architecture for evaluating clauses in parallel. In their architecture, the clauses are separated into a number of groups and the deduction is performed in parallel. Then the results are merged together to allow the assignment to the variables.

A software/hardware solver for SAT was introduced in [30]. In their approach, they minimized the hardware compilation time which greatly reduced the total time to solve the problem. They also implemented their solver on an FPGA.

III. Routing and Wavelength Assignment in Optical Networks

In optical networks, wavelength routers perform the function of switches in non-optical networks. Figure 1 shows a wavelength router with three input links and three output links. Each link uses three wavelengths $\lambda_1, \lambda_2, \lambda_3$. The function of a router is to switch packets from input links to output links. Depending on the switch type, it might or might not be able of doing wavelength conversion.

Simple switches do not perform wavelength conversion, meaning that if a packet is arriving at input I_1, it could be switched to any output link, however it must use the same wavelength it arrived with on I_1. That makes the switch design simple, however, it can not fully utilize the network resources. For example if a node arrives at I_1 using wavelength λ_1, and the best path to the destination is edge O_3, However, λ_1 is used by another connection at O_3. In this case, the connection must be routed through another edge and node.

Advanced switches can perform wavelength conversion between input and output links [25], thus a packet can travel on different wavelengths on different links from the source to the destination. In this case these switches can fully utilize the network resources on the expense of cost and complexity. With fixed routing, where the path between any two nodes is calculated and set up based on shortest path, it has been reported that wavelength conversion results in 30%-40% improvement in blocking probability compared with no wavelength conversion [19].

Establishing a path from a source node to a destination node involves determining the path and the wavelength assignment on every link on the path. These two problems could be solved independently (first we find the path, then we perform wavelength assignment on the chosen links) or we can solve these two problems jointly thus producing a better solution with a more complicated algorithm.

Routing in optical networks has been the subject of intensive research [8] and was analyzed in [26]. Adaptive routing was reported in [21]. Three routing strategies were compared in [16] with respect to performance and the size of the network, while alternate routing has been studied in [9].

In this paper we are interested in using advanced SAT solvers to solve the problem of routing and wavelength assignment in optical networks. To illustrate our approach, lets consider the network in Figure 2. In the figure, each node is labeled by an upper-case letter, and each link is marked by (x, n) where x is the name of the link and n is a positive integer that represents the weight, i.e. cost, of the link. We will assume that each edge has w wavelengths and the variable a_z denotes wavelength z in edge a. Node I and H are the *source* and *destination* nodes, respectively. The objective is to find a path from I to H that minimizes the total path cost (sum of weights of all links in the path).

Two sets of *variables* are defined for the problem:

- A Boolean variable is defined for each node. A value of 1 (0) for each variable indicates that the corresponding node is (is not) included in the optimal path from the source node to the destination node.
- A Boolean variable is defined for each wavelength in every edge. A value of 1 (0) for each variable indicates that the corresponding wavelength is (is not) included in the optimal path from the source node to the destination node. Initially, we set x_z to zero, if wavelength z in link x is being used by another connection.

In the above example, if we assume 4 wavelengths per link, a total of 57 variables are declared, 9 of which represent the nodes $(A, B, C, \ldots, D)$ and 48 variables represent the edges with their 4 wavelengths $(a_1, a_2, a_3, a_4, b_1, \ldots l_4)$. The following set of *constraints* are generated:

- For both source and destination nodes, only one of the wavelengths in the neighboring edges will be part of the path. This can be expressed using the following two PB constraints for the above example:

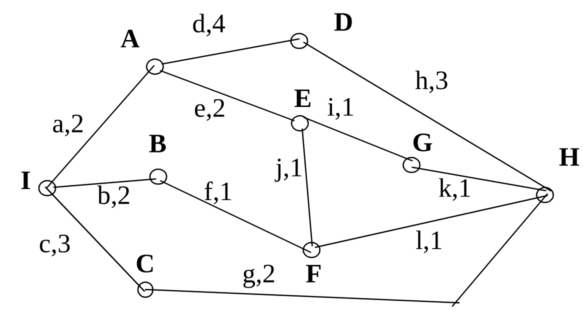

Fig. 2. An example of a network with 9 nodes and 12 edges. Upper-case letters represent nodes. Lower-case letters represent edges. Each edge is associated with an integer representing its weight.

$$\sum_{i=0}^{w-1} a_i + \sum_{i=0}^{w-1} b_i + \sum_{i=0}^{w-1} c_i = 1 \quad (4)$$

$$\sum_{i=0}^{w-1} h_i + \sum_{i=0}^{w-1} k_i + \sum_{i=0}^{w-1} l_i + \sum_{i=0}^{w-1} g_i = 1 \quad (5)$$

- All other nodes (except the source and destination nodes) will either be (i) part of the path or (ii) not part of the path. In the first case, *exactly* two wavelengths in two of the edges connected to that node will be part of the path. In the second case, none of the wavelengths in any of the edges connected to the node will be part of the path. This can be expressed using a single PB constraint for each node. In the above example the PB expression for node A is as follows:

$$2\bar{A} + \sum_{i=0}^{w-1} a_i + \sum_{i=0}^{w-1} d_i + \sum_{i=0}^{w-1} e_i = 2 \quad (6)$$

If node A is within the path, then variable A is true. Hence $\bar{A} = 0$ and the only way to satisfy the expression is to set two wavelengths in the neighboring edges to true, i.e. making them part of the path. If node A is not within the path, then variable A is set to *0*. Hence $\bar{A} = 1$ and the only way to satisfy this expression is to set all three variables to 0, i.e. none of the edges are part of the path. Similar PB constraints are generated for the other nodes, for example, node B gets:

$$2\bar{B} + \sum_{i=0}^{w-1} b_i + \sum_{i=0}^{w-1} f_i = 2 \quad (7)$$

(8)

To ensure that no two wavelengths from the same edge are used in the same optimal path, we add the following new constraint for each edge *z:*

$$\sum_{i=0}^{w-1} z_i \le 1 \qquad \forall z \in Edges \quad (9)$$

That will ensure a maximum of only one chosen wavelength in any link. If no wavelengths are chosen in a particular link, then the link is not part of the optimal path.

For the switches without wavelength conversion, we add w constraints, i.e. equal to the number of wavelengths per edge, for each available node forcing the optimal path to use the same wavelength among all edges. For example, the constraint for node A is:

$$a_i \oplus d_i \oplus e_i = 0 \qquad \forall i \in Wavelengths \quad (10)$$

this can be expressed using 4 CNF constraints as following:

$$(\bar{a}_i \vee \bar{d}_i \vee \bar{e}_i) \wedge (\bar{a}_i \vee d_i \vee e_i) \wedge \\ (a_i \vee \bar{d}_i \vee e_i) \wedge (a_i \vee d_i \vee \bar{e}_i) \quad (11)$$

Note that the 4 CNF constraints prohibit the 4 assignments: 111, 100, 010, and 001 for variables *a,d,e,* respectively. In other words, it disallows the selection of an odd number of identical wavelengths in each node. The first CNF constraint in (11) can also be discarded, since it is impossible to enable 3 wavelengths per node for the same optimal path (violates constraint (9)).

The above set of constraints guarantee that a complete path will be formulated from the source node to the destination node. To minimize the total cost of the path, a PB objective function, consisting of all wavelength variables with their corresponding edge weights, is created as follows:

$$min\left(\begin{array}{l} 2\left(\sum_{z=0}^{w-1} a_z\right) + 2\left(\sum_{z=0}^{w-1} b_z\right) + 3\left(\sum_{z=0}^{w-1} c_z\right) + 4\left(\sum_{z=0}^{w-1} d_z\right) + \\ 2\left(\sum_{z=0}^{w-1} e_z\right) + f + 2\left(\sum_{z=0}^{w-1} g_z\right) + 3\left(\sum_{z=0}^{w-1} h_z\right) + i + j + k + l \end{array}\right) \quad (12)$$

In general, the minimization could be represented as

$$minimize\left(\sum_{\forall a} weight_a \times var_{ai}\right) \quad (13)$$

where $weight_a$ and var_{ai} represent the cost and variable of wavelength i in edge a, respectively. Other objective functions can also be expressed. For example if the goal is to reduce the number of nodes in the path, the objective function in can be replaced by

$$min(A + B + C + D + E + F + G + H + I) \quad (14)$$

which only consists of the sum of node variables without taking into consideration the edge variables or the edge weights.

By formulating the problem as such, we can do more than finding the minimum cost path. We can incorporate any restrictions that we can think of in the resulting path. For example, by adding the PB constraint $A = 1$, we are forcing node *A* to be part of the minimal cost path. Similarly, we can exclude node *A* from the solution by adding the PB constraint $A = 0$.

We can also add dependencies between nodes. For example, we can force one of two nodes, e.g. *J* and *B*, to exist in the resulting path. This can be expressed by adding the following two CNF constraints:

$$(J \vee B)(\bar{B} \vee \bar{J}) \quad (15)$$

We can also force certain nodes to be in the path only if a specific node is. For example, we can force nodes *B*, *C*, and *D* to be part of the solution if and only if node *A* is. This is expressed using the following set of CNF constraints:

$$(\bar{A} \vee B)(\bar{A} \vee C)(\bar{A} \vee D)(A \vee \bar{B})(A \vee \bar{C})(A \vee \bar{D}) \quad (16)$$

Note that the complexity of converting the graph into a SAT problem is $O(v + e + k)$, where *v* is the number of nodes, *e* is the number of edges, and *k* is the number of graph restrictions (e.g. (15) and (16)).

IV. Experimental Results

In this section, we evaluate the use of SAT solvers in identifying the shortest routing path and wavelength assignments (RWA) in optical networks. The RWA problem was encoded as a SAT instance as shown in Section III. Topology generation has been an active area of research. Therefore, we decided to use the BRITE topology generator [6] to produce different random topologies to test our approach. BRITE can produce multiple generation models and can assign links attributes such as bandwidth and delay.

We created networks of different sizes with the number of nodes ranging from 20-50. The number of links per node was set to 2 and the number of wavelengths per edge ranged from 5-10. Nodes are placed randomly in a plane with a side of 1000 units. We considered the weight of the link as the Euclidean length of the link (we can choose any weight factor but choose the distance since it is already generated by the topology simulator). The topology model is Waxman model [32] with parameters $\alpha = 0.15, \beta = 0.2$. The goal was to find the path with the minimal edge cost.

The network is stored in a text file and passed to a PERL script that converts it into a SAT-encoded problem. The SAT problem is then solved by advanced SAT solvers. For our experiments, we used the PBS [2, 4] and MiniSAT [15] solvers. PBS and MiniSAT are new solvers than can handle both CNF and PB constraints and can solve *decision* and *optimization* problems. Both implement the latest enhancements in the SAT domain. Both have won several awards in the annual PB-SAT Competition [28]. PBS can also solve optimization problems using a *linear*-based or *binary*-based search schemes (MiniSAT uses a default *linear*-based search scheme). Both schemes have shown competitive performance on various optimization instances that consists of CNF-only or CNF/PB constraints. The experiments were conducted on a Pentium Xeon 3.2 Ghz machine, equipped with 4 GBytes of RAM, and running Linux. The runtime limit was set to 1000 seconds.

Table 1 lists the runtime results for the optical routing benchmarks. The table lists the name of the instance, the runtime in seconds of PBS using linear-based and binary-based search schemes, the status of the search (whether the instance is satisfiable or unsatisfiable), and the size of the shortest path if the instance is satisfiable. The name of the instances of the form *X_Y_A_B_wC_dD* indicates that the instance has *X* nodes and the number of links per node is *Y* and the number of wavelengths per link is *C*. The randomly selected source and destinations nodes are *A* and *B*, respectively. Since the time to find the optimal path depends on the network load, we simulated a busy network by randomly disabling various wavelengths and measuring the corresponding solver time needed to find the optimal path. The percentage of randomly disabled wavelengths (i.e. the higher the percentage, the more loaded is the network) is *D*. Several observations are in order:

- PBS and MiniSAT were able to identify the shortest path in all reported cases.
- Binary search seems to be more competitive than the linear search, especially for the larger instances consisting of 50 nodes.
- As more wavelengths are disabled (i.e. used by other paths), the cost of the optimal path increases. For example, for the 20_2_17_12_w5 instance, the cost of the optimal path was 700 when disabling 50% of the wavelengths, but that increased to 1130 when disabling 60% of the wavelengths. Interestingly, the problem becomes easier to solve, i.e. lower search runtimes, as more wavelengths are disabled since the solver has less choices to make or search through.
- All problems became unsatisfiable, i.e. no paths were available, once 80% of the wavelengths were randomly disabled.
- The larger the network grid, the longer is the search runtime.
- The approach is complete and is guaranteed to find the shortest path given enough time and memory resources. Even if the solver times-out, it will return the shortest discovered path.

TABLE 1. Experimental results on various size optical network grids using the PBS4 and MiniSAT Solvers. (S. Path = Shortest Path. S/U = Satisfiable or Unsatisfiable)

Instance Name	PBS4						MiniSAT		
	Binary Search			Linear Search			Linear Search		
	Time	S/U	S. Path	Time	S/U	S. Path	Time	S/U	S. Path
20_2_17_12_w5_d0	0.04	S	700	0.06	S	700	5.79	S	700
20_2_17_12_w5_d10	0.03	S	700	0.04	S	700	1.59	S	700
20_2_17_12_w5_d20	0.03	S	700	0.04	S	700	3.03	S	700
20_2_17_12_w5_d30	0.02	S	700	0.02	S	700	0.9	S	700
20_2_17_12_w5_d40	0.01	S	700	0.01	S	700	0.36	S	700
20_2_17_12_w5_d50	0	S	700	0	S	700	0.34	S	700
20_2_17_12_w5_d60	0	S	1130	0	S	1130	0.04	S	1130
20_2_17_12_w5_d70	0	S	1373	0	S	1373	0.01	S	1373
20_2_17_12_w5_d80	0	U		0	U		0.01	U	
20_2_17_12_w10_d0	0.3	S	700	0.29	S	700	26.6	S	700
20_2_17_12_w10_d10	0.5	S	700	0.59	S	700	7.65	S	700
20_2_17_12_w10_d20	0.05	S	700	0.05	S	700	10.1	S	700
20_2_17_12_w10_d30	0.07	S	700	0.08	S	700	4.76	S	700
20_2_17_12_w10_d40	0.03	S	700	0.03	S	700	2.2	S	700
20_2_17_12_w10_d50	0.01	S	700	0.01	S	700	0.68	S	700
20_2_17_12_w10_d60	0	S	700	0	S	700	0.33	S	700
20_2_17_12_w10_d70	0.01	S	1023	0.01	S	1023	0.01	S	1023
20_2_17_12_w10_d80	0	U		0	U		0.01	U	
50_2_8_45_w5_d0	0.08	S	309	7.29	S	309	175.8	S	309
50_2_8_45_w5_d10	0.08	S	309	3.05	S	309	102.2	S	309
50_2_8_45_w5_d20	0.25	S	309	1.13	S	309	239	S	309
50_2_8_45_w5_d30	0.05	S	309	0.55	S	309	4.23	S	309
50_2_8_45_w5_d40	0.05	S	309	0.23	S	309	8.88	S	309
50_2_8_45_w5_d50	0.01	S	309	0.02	S	309	2.23	S	309
50_2_8_45_w5_d60	0.01	S	2760	0.01	S	2760	0.73	S	2760
50_2_8_45_w5_d70	0	U		0	U		0.11	U	
50_2_8_45_w10_d0	0.22	S	309	10.1	S	309	0.02	S	309
50_2_8_45_w10_d10	0.34	S	309	5.56	S	309	2.97	S	309
50_2_8_45_w10_d20	0.21	S	309	1.18	S	309	2.21	S	309
50_2_8_45_w10_d30	0.1	S	309	1.14	S	309	6.17	S	309
50_2_8_45_w10_d40	0.13	S	309	0.81	S	309	10.5	S	309
50_2_8_45_w10_d50	0.15	S	309	0.26	S	309	3.64	S	309
50_2_8_45_w10_d60	0.18	S	2365	0.03	S	2365	2.07	S	2365
50_2_8_45_w10_d70	0.01	S	2365	0.01	S	2365	0.22	S	2365
50_2_8_45_w10_d80	0	U		0	U		0.01	U	

V. CONCLUSION

In this paper, we presented a new approach to solving the routing and wavelength assignment problem in optical networks using Boolean Satisfiability (SAT) solvers. We showed how to formulate the RWA problem as a SAT instance and evaluated several advanced SAT techniques. The approach was tested on a number of networks of various sizes and showed promising results. The presented approach is *complete* and will find the *shortest* possible path. One of the advantages of the new approach is the ability to add user-specific constraints that can restrict the existence of certain nodes and edges in the resulting path.

REFERENCES

[1] M. Abramovici, and D. Saab "Satisfiability on Reconfigurable Hardware," in *Proc. of the Int'l Workshop on Field Programmable Logic and Application*, 448-456, 1997.

[2] F. Aloul, A. Ramani, I. Markov, and K. Sakallah, "Generic ILP Versus Specialized 0-1 ILP: An Update," in *Proc. of ICCAD*, 450-457, 2002.

[3] F. Aloul, S. Hassoun, K. Sakallah, and D. Blaauw, "Robust SAT-Based Search Algorithm for Leakage Power Reduction," in *Proc. of PATMOS*, 167-177, 2002.

[4] B. Al-Rawi and F. Aloul, PBS4 SAT Solver, 2005. Available at: *http://www.eecs.umich.edu/~faloul/Tools/pbs4*

[5] A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu, "Symbolic Model Checking using SAT procedures instead of BDDs," in *Proc. of Design Automation Conference*, 317-320, 1999.

[6] BRITE Topology Generator, 2005. *http://www.cs.bu.edu/brite*

[7] D. Chai and A. Kuehlmann, "A Fast Pseudo-Boolean Constraint Solver", in *Proc. of Design Automation Conference* (DAC), 830-835, 2003.

[8] I. Chlamatac, A. Ganz, and G. Karmi "Purely Optical networks for Terabit Communications," in *Proc. IEEE INFOCOM'89*, 887-896, 1989.

[9] S. Chung, A. Kashper, and K. Ross "Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing," in *Proc. IEEE/ACM Trans. on Networking*, vol. 1, 105-115, Feb. 1993.

[10] S. A. Cook, "The Complexity of Theorem Proving Procedures," in *Proc. of the ACM Symposium on the Theory of Computing*, 151-158, 2004.

[11] N. Creignou, S. Kanna, and M. Sudan, "Complexity Classifications of Boolean Constraint Satisfaction Problems", *SIAM Press*, 2001.

[12] A. Dandalis, and V. K. Prasanna, "Run-time Performance Optimization of an FPGA Based Deduction Engine for SAT Solvers," in *ACM TODAES*, 7(4), 547-562, October 2002.

[13] M. Davis, G. Longman, and D. Loveland "A Machine Program for Theorem Proving," in *J. of the ACM*, 5(7), 394-397, 1962.

[14] H. Dixon and M. Ginsberg, "Inference Methods for a Pseudo-Boolean Satisfiability Solver," in *Proc. of* AAAI, 635-640, 2002.

[15] N. Een and N. Sorensson, "An Extensible SAT-solver," in *Proc. of the SAT*, 502-508, 2003.

[16] M. Garnot, M. Sotom, and F. Masetti, "Routing Strategies for Optical Paths in WDM Networks," in *Proc. of* ICC, 422-426, June 1997.

[17] E. Goldberg and Y. Novikov, "BerkMin: A Fast and Robust SAT-solver," in *Proc. of DATE*, 142-149, 2002.

[18] J. Hayes, "Introduction to Digital Logic Design," *Addison-Wesley*, 1993.

[19] M. Kovacevic and A. Acampora, "Benefits of Wavelength Translation in All-Optical Clear-Channel Networks" in *IEEE J. of Selected Areas in Comm*. vol. 14, 868-888, June 1996.

[20] J. Marques-Silva and K. Sakallah "GRASP: A search algorithm for propositional satisfiability," in *IEEE Trans. on Comp.*, 48(5), 506-521, 1999.

[21] A. Mokhtar and M. Aizioglu, "Adaptive Wavelength Routing in All-Optical Networks," in *IEEE Trans. on Networking*, v. 6, 197-206, 1996.

[22] K. Moskewicz, C. Madigan, Y. Zho, and S. malik "Chaff: Engineering an efficient SAT solver," in *Proc. of* DAC, 503-535, 2001.

[23] G. Nam, F. A. Aloul, K. A. Sakallah, and R. Rutenbar, "A Comparative Study of Two Boolean Formulations of FPGA Detailed Routing Constraints," in *Proc. of* ISPD, 222-227, 2001.

[24] S. Personick, "Evolving toward the Next-Generation Internet: Challenges in the Path Forward," in *IEEE Communication*s, 40(7), 72-76, 2002.

[25] B. Ramamurthy and B. Mukherjee, "Wavelength Conversion in WDM Networking," in *IEEE J. Select Areas in Comm.*, v. 16, 1061-1073, 1998.

[26] R. Ramaswami and K. Sivarajan. "Routing and Wavelength Assignment in all-Optical Networks," in *IEEE/ACM Trans. on Networking*, v. 3, 858-867, October 1996.

[27] R. Ramaswami and K. Sivarajan, "Optical Networks: A Practical Perspective," *Morgan Kaufmann*, 2002.

[28] SAT Competition. *http://www.satcompetition.org*

[29] H. Sheini and K. Sakallah, "Pueblo: A Modern Pseudo-Boolean SAT Solver," in *Proc. of* DATE, v. 2, 684-685, 2005.

[30] I. Skliarova and A. Ferrari "A Software/Reconfigurable Hardware SAT Solver," in *IEEE Trans. on Very Large Scale Systems*, 12(4), 408-419, April 2004.

[31] N. Skorin-Kapov, "Heuristic Algorithms for the Routing and Wavelength Assignment of Scheduled Lightpaths Demands in Optical Networks," in *IEEE J. on Selected Areas in Communication*, 24(8), August 2006.

[32] B. Waxman, "Routing of Multipoint Connections," in *IEEE J. of Selected Areas in Comm.*, 6(9), 1617-1622, December 1988.

[33] P. Zhong, M. Martonosi, P. Ashar, and S. Malik, "Using Reconfigurable Computing to Accelerate Boolean Satisfiability," in *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 18(6), 861-868, 1999.