Routing and wavelength assignment by partition coloring

Thiago F. de Noronha[*] Celso C. Ribeiro[*]

[*]Department of Computer Science
Catholic University of Rio de Janeiro
Rio de Janeiro, RJ 22453-900, Brazil

E-mails: {tfn,celso}@inf.puc-rio.br

Abstract

We show in this work how the problem of routing and wavelength assignment in all-optical networks may be solved by a combined approach involving the computation of alternative routes for the lightpaths, followed by the solution of a partition coloring problem in a conflict graph. A new tabu search heuristic is also proposed for the partition coloring problem, which may be viewed as an extension of the graph coloring problem. Computational experiments are reported, showing that the tabu search heuristic outperforms the best heuristic for partition coloring by approximately 20% and illustrating that the new approach for the problem of routing and wavelength assignment is more robust than a well established heuristic for this problem.

Keywords: Routing, wavelength assignment, optical networks, WDM, partition coloring, heuristics, tabu search.

1 Motivation

The technology of Optical Networks is increasingly important in telecommunications and appears in a large number of applications such as video-conferences, high performance computing, and grids, among others. The information is transmitted along the optical fibers as optical signals. Each link operates at a speed of the order of terabits per second, which is much faster than the currently available electronic devices for signal reception and transmission. Wavelength Division Multiplexing (WDM) allows more efficient use of the huge capacity of optical fibers, as far as it permits the simultaneous transmission of different channels along the same fiber, each of them using a different wavelength.

In all-optical networks, each signal is converted to the optical domain and reaches the receptor without conversion to the electrical domain. Keeping the signal at the optical domain allows higher transmission speeds, eliminating the overheads imposed by electrical conversions. Such networks require a large number of available wavelengths, especially when wavelength conversion is not possible in the nodes. Routing should be performed so as to minimize the number of wavelengths needed.

We consider all-optical networks in which the nodes make use of wavelength-routing switches. Each signal is routed by an optical switch according to its input port and the wavelength used. Each wavelength at each input port is mapped to a different output port. An all-optical point-to-point connection between two endnodes is called a lightpath. Two lightpaths may use the same wavelength, provided they do not share any common link. In this work, we assume that wavelength conversion along a lightpath is not permitted, since this technology is not yet fully available nowadays. Therefore, each lightpath should use the same wavelength from the transmitter to the receiver.

The Routing and Wavelength Assignment (RWA) problem consists in routing a set of lightpaths and assigning a wavelength to each of them. Variants of RWA are characterized by different optimization criteria and traffic patterns, see e.g. [5, 14]. We consider in this work the min-RWA offline variant, in which all connection requirements are known beforehand and one seeks to minimize the total number of wavelengths used for routing these connections. Erlebach and Jansen [6] showed that min-RWA is NP-hard. Several authors proposed different approximate algorithms for solving min-RWA. Bannerjee and Mukherjee [3], Hyytiä and Virtamo [9], and Li e Simha [11] decompose the problem in two phases. In the first phase one or more possible routes are computed for each lightpath, while in the second one route and one wavelength are assigned to each lightpath. The second phase is often solved as a coloring problem defined on a conflict graph. Manohar, Manjunath, and Shevgaonkar [12] proposed the Greedy-EDP-RWA construction which was used in a multistart procedure heuristic, in which both subproblems are solved simultaneously. Their algorithm is much faster and finds solutions as good as those found by the algorithm in [3].

We propose a new two-phase heuristic for min-RWA, using the same decomposition scheme described in [11]. In the first phase, a set of possible routes is precomputed for each lightpath. In the second, a route (among those precomputed) and a wavelength are assigned for each lightpath by solving a partition coloring problem, which is formulated in the next section. A tabu search for solving the partition coloring problem is proposed in Section 2. The complete heuristic for min-RWA is presented in Section 3. Computational results comparing the tabu search procedure for partition coloring and the new heuristic for min-RWA with other approaches in the literature are reported in Section 4. Concluding remarks are drawn in the last section.

2 A Tabu Search Heuristic to the Partition Coloring Problem

2.1 Partition Coloring Problem

Let $G = (V, E)$ be a non-directed graph, where E is the set of edges and V is the set of nodes. Let also $V_1, V_2, \ldots, V_q$ be a partition of V into q subsets such that $V_1 \cup V_2 \cup \ldots \cup V_q = V$ and $V_i \cap V_j = \emptyset, \forall i, j = 1, \ldots, q$ with $i \neq j$. We refer to $V_1, V_2, \ldots, V_q$ as the *components* of the partition. The *Partition Coloring Problem* (PCP) consists in finding a subset $V' \subset V$ such that $|V' \cap V_i| = 1, \forall i = 1, \ldots, q$ (i.e., V' contains one node from each component $V_i, \forall i = 1, \ldots, q$), and the chromatic number of the graph induced in G by V' is minimum. This problem is clearly a generalization of the graph coloring problem. Li and Simha [11] have shown that the decision version of PCP is NP-complete.

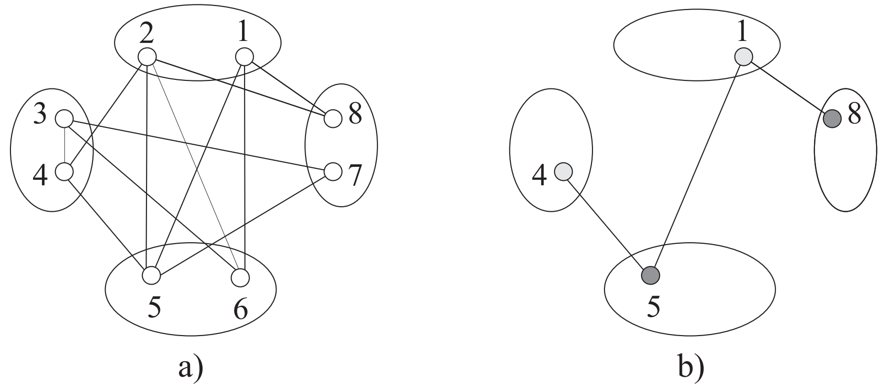

Figure 1: (a) Instance of PCP and (b) optimal solution with two colors.

We illustrate an instance of PCP in Figure 1. The associated graph has eight nodes and four components. The optimal solution makes use of two colors: the first color is used to color nodes 1 and 4, while the second is used to color nodes 5 and 8.

2.2 Construction Heuristic

Li and Simha [11] proposed six construction algorithms for PCP, based on graph coloring heuristics. Best results were obtained with algorithm `onestepCD` (One Step Color Degree), which is also used to build initial solutions to our tabu search procedure. Algorithm `onestepCD` described in Figure 2 is a variant of the color degree heuristic [4] adapted to PCP. We denote by $c(i)$ the component of node $i \in V$ (i.e. $i \in V_{c(i)}$) and by $V' \subseteq V$ the set of nodes of G already colored in a partial solution under construction to PCP. The color saturation degree $CD(i)$ of each node $i \in V$ is defined as the number of different colors used to color the neighbors of i in V'.

procedure `onestepCD`(G)
1. Remove from G all edges $(i, j) \in E : i, j \in V^k$ for some $k = 1, \ldots, q$;
2. Set $V' \leftarrow \emptyset$;
3. **while** $|V'| < q$ **do**
4. Set $X \leftarrow \emptyset$;
5. **for** $k = 1, \ldots, q : V_k \cap V' = \emptyset$ **do**
6. Set $X \leftarrow X \cup \text{argmin}\{CD(i) : i \in V^k\}$;
7. **end-for**;
8. Set $x \leftarrow \text{argmax}\{CD(i) : i \in X\}$;
9. Set $V' \leftarrow V' \cup \{x\}$;
10. Assign the minimum possible color to x;
11. Remove from G all nodes in $V_{c(x)} \setminus \{x\}$;

12 **end-while**;

end `onestepCD`.

Figure 2: Pseudo-code of the `onestepCD` construction algorithm to PCP.

In line 1, all edges connecting nodes in the same component of the partition are removed.

The set of nodes already colored is initialized in line 2. The loop in lines 3-12 is performed until all partitions have been colored. In lines 4-7, we build the set X formed by the nodes with least colored saturation degree from each uncolored component. The node x with the maximum colored saturation degree among those in X is selected in 8. Node x is colored with the minimum possible color in lines 9 and 10. All uncolored nodes in the same partition as x are removed from the graph in line 11 and the above steps are repeated. The overall complexity of the algorithm `onestepCD` is $O(m + qn)$.

2.3 Tabu Search for PCP

Algorithm `onestepCD` is applied to create a feasible initial solution S to PCP with $\texttt{max}C$ colors. The tabu search [8] procedure `TS-PCP` described in Figure 3 is based on a first-improving local search strategy using a 1-opt neighborhood and attempts to find a feasible solution with a smaller number of colors. We define a *coloring conflict* as a pair of adjacent nodes in different components of the graph which are colored with the same color. We denote by $\texttt{conflicts}(S)$ the number of coloring conflicts in any solution S.

A new possibly infeasible solution S' using $\texttt{max}C - 1$ colors is built in line 1. The iteration counter and the tabu list are initialized in line 2. The set Q of components involved in coloring conflicts is built in line 3. The loop in lines 4-18 investigates the neighborhood of the current solution until no further reduction in the number of coloring conflicts is possible. Each neighbor is obtained by (a) recoloring with a different color exactly one node involved in a coloring conflict or (b) changing (and coloring) the node that is colored in a component involved in a coloring conflict. A component $k \in Q$ involved in a coloring conflict is randomly selected in line 5. Variable *reduction* is initialized as `.FALSE.` in line 6 and will be used to indicate that an improving solution with a smaller number of conflicts was found. The number `maxConflicts` of coloring conflicts in the neighbor of the current solution with the least number of conflicts is initialized in line 7. The loop in lines 8-17 investigates all neighbors corresponding to the selected component k. The test in line 9 discards all neighbors which belong to the tabu list and do not satisfy an aspiration criterion. A new tentative solution S'' is built in line 11 by coloring a node $i \in V^k$ with a color $\ell = 1, \ldots, \texttt{max}C - 1$. The best neighbor $\bar{S}$ of the current solution is updated in lines 12-13. If the number of coloring conflicts in S'' is smaller than those in S', then the latter is updated in line 15, together with the set of components involved in coloring conflicts and the flag *reduction* that indicates that an improving solution was found. The iteration counter is also updated in line 15.

If there are no coloring conflicts in the new solution S', the current solution S using fewer colors is updated in lines 21 and 22, and a new attempt to reduce the number of colors is started in line 23. Otherwise, only a non-improving neighbor $\bar{S}$ of the current solution S' was found. We identify in line 25 the node $\bar{i}$ colored with color $\bar{\ell}$ in the current solution S'. The pair $\bar{i}$, $\bar{\ell}$ is inserted in the tabu list for `TabuTenure` iterations in line 26. In line 27, the current solution S' moves to the best neighbor $\bar{S}$. The iteration counter is updated in line 28. If the stopping criterion is not yet satisfied, a new iteration resumes in line 29.

The maximum number of iterations used as the stopping criterion is set as $\texttt{maxIter} = q \cdot (\texttt{max}C - 1) \cdot F_{end}$, where q is the number of components in the graph, $\texttt{max}C - 1$ is the tentative number of colors, and F_{end} is a parameter to be tuned, as described in Section 4.1.

```
procedure TS-PCP(G, S, maxC)
1.  Build a new solution S' by randomly recoloring with any of the colors 1, ..., maxC − 1
        every node originally colored with color maxC;
2.  Free the tabu list and set iter ← 0;
3.  Let Q be the set formed by all components involved in color conflicts in S';
4.  while Q ≠ ∅ do
5.      Randomly select k ∈ Q and update Q ← Q \ {k};
6.      Set reduction ← .FALSE.;
7.      Set maxConflicts ← ∞;
8.      for each i ∈ V_k and for each color ℓ = 1, ..., maxC − 1 while .NOT.reduction do
9.          if the pair i, ℓ is not in the tabu list or if it satisfies the aspiration criterion
10.         then do;
11.             Obtain a tentative solution S'' by recoloring node i with color ℓ;
12.             if conflicts(S'') < maxConflicts
13.             then maxConflicts ← conflicts(S''), S̄ ← S'', and k̄ ← k;
14.             if conflicts(S'') < conflicts(S')
15.             then S' ← S'', reduction ← .TRUE., iter ← iter + 1, and update Q;
16.         end-if;
17.     end-for;
18. end-while;
19. if conflicts(S') = 0
20. then do;
21.     S ← S';
22.     maxC ← maxC − 1;
23.     return to step 1;
24. else do
25.     Let ī be the node currently colored with color ℓ̄ in component V_k̄ of S';
26.     Insert the pair ī, ℓ̄ in the tabu list for TabuTenure iterations;
27.     S' ← S̄;
28.     iter ← iter + 1;
29.     if iter < maxIter then return to step 3;
30. end-if;
31. return S;
end TS-PCP.
```

Figure 3: Pseudo-code of the TS-PCP local search algorithm to PCP.

3 A New Heuristic to min-RWA

Let $N = (X, A)$ be a graph representing the physical topology of an all-optical-network, where X is the set of switches (nodes) and A is the set of links (edges) connecting the switches. Furthermore, we denote by T the set of connections to be established. Each connection is defined by a pairs of endnodes in X.

In this section we describe a new two-phase heuristic for min-RWA. First, we compute a set of alternonestepCDative possible routes for each connection using the k-EDR construction procedure sumarized below. Next, we build a conflict graph, in which the nodes correspond to routes and there is an edge between each pair of nodes whose associated routes share a common link. All alternative routes associated to the same connection are placed in the same component of the conflict graph. In the second phase, the tabu search heuristic TS-PCP is applied to solve the above defined instance of PCP. The colored nodes define the routes and the colors define the selected wavelengths.

The pseudo-code of the construction procedure k-`EDR` is given in Figure 4. Its main component is the heuristic `BGAforEDP` proposed by Kleinberg [10] for the *maximum edge disjoint path problem*, which finds the largest subset of connections that can be routed by edge-disjoint paths with a maximum number d of edges. We notice that routing with as many disjoint paths as possible allows a better use of the available wavelengths. The maximum number d of edges in each route is initialized in line 1 as suggested in [10]. The set R of routes is initialized in line 2. The loop in lines 3-19 attempts to build at most k alternative routes for each connection in T. A copy T' of the set of connections to be established is made in line 4. The loop in lines 5-18 computes a single route for each connection. A copy N' of the network is made in line 6. In line 7 we build a list L defining a random permutation of the order in which the connections still in T' will be scanned. The loop in lines 8-17 finds a set of disjoint routes with at most d edges for as many connections in T' as possible. The next connection t to be investigated is selected in line 9. The shortest path s in the network N' between the endnodes of connection t is computed in line 10. Line 11 checks if this path has at most d edges. In this case, the set T' is updated in line 13, the set of routes is updated in line 14, and the residual network N' is upated in line 15. The set R of routes is returned in line 20.

procedure k-`EDR`(N, T, k)
1. Set $d \leftarrow max(diameter(N), \sqrt{|A|})$;
2. Set $R \leftarrow \emptyset$;
3. **for** $j = 1, \ldots, k$ **do**
4.   Set $T' \leftarrow T$;
5.   **while** $T' \neq \emptyset$ **do**
6.   Set $N' \leftarrow N$;
7.   Build a list L with a random permutation of the connections in T';
8.   **for** $i = 1, ..., |T'|$ **do**
9.   Let t be the i-th connection in L;
10.   Compute the shortest path s in N' between the endnodes of t;
11.   **if** s has no more than d edges
12.   **then do**
13. $T' \leftarrow T' \setminus \{t\}$;
14. $R \leftarrow R \cup \{(t, s)\}$;
15.   Remove the edges in path s from N';
16.   **end-if**;
17.   **end-for**;
18.   **end-while**
19. **end-for**;
20. **return** R;

end k-`EDR`

Figure 4: Pseudo-code of the k-`EDR` construction heuristic.

4 Computational Results

Algorithms `onestepCD`, `TS-PCP`, and k-`EDR` described in the previous sections, as well as the heuristic proposed by Manohar, Manjunath, and Shevgaonkar [12] for min-RWA, were implemented in C++ and compiled with version v2.96 of the Linux/GNU compiler. All experiments were performed on a Pentium IV machine with a 2.0 GHz clock and 512 Mbytes of RAM memory. Numerical results concerning the partition coloring problem are reported in Section 4.1.

Computational results regarding the heuristics for min-RWA are reported in Section 4.2.

4.1 Tabu Search for PCP

For the first experiment, we generated a small set of four PCP instances with 500 components each. These instances were built from node coloring instances DJSC250.5 and DJSC500.5 [2] available from `http://mat.gsia.cmu.edu/COLOR/instances.html`. They are used exclusively for tuning once for all the parameters of the tabu search heuristic `TS-PCP`. PCP instance DJSC500.5-1 has exactly one node in each component and coincides with the original DJSC500.5 vertex coloring instance. Further PCP instances were generated by adding 1, 2, or 3 new nodes to each component of instance DJSC500.5-1. Edges adjacent to the new nodes were randomly generated with the same probability 0.5, as in the original node coloring instance. The main characteristics of these instances are summarized in Table 1.

Instance	Nodes	Components	Nodes per component	Density
DJSC500.5-1	500	500	1	0.5
DJSC500.5-2	1000	500	2	0.5
DJSC500.5-3	1500	500	3	0.5
DJSC500.5-4	2000	500	4	0.5

Table 1: Characteristics of the PCP instances used for tuning the tabu search heuristic.

The tabu search heuristic `TS-PCP` was applied to instances DJSC500.5-1 to DJSC500.5-4 with different values for the parameters `TabuTenure` and F_{end} which defines the stopping criterion. Numerical results are reported in Tables 2 a 5. For each combination of parameters, we report the minimum, average, and maximum number of colors over 10 runs using different seeds for each instance. Best average results are underlined and were always obtained with the tabu tenure parameter randomly generated from a uniform distribution in the interval $[0, \frac{1}{2}C']$, with $C' = \mathtt{max}C - 1$. The parameter $F_{end} = 5$ was set at the intermediary value. This combination of parameters will be used in all remaining experiments.

In the next experiment we investigate the behavior of heuristics `onestepCD` and `TS-PCP` (with the parameter values previously set) for randomly generated graphs with 100 to 900 components. Each component has exactly two nodes and the edge density is 0.5. Ten instances were generated for each number of nodes and algorithms `onestepCD` and `TS-PCP` were run 10 times with different seeds. The average number of colors found by each heuristic over 100 runs for each graph size is plotted in Figure 5. We observe that for both algorithms the average number of colors increases linearly with the number of components. For the largest graphs with 900 nodes, the number of colors used by the new tabu search heuristic is approximately 80% of that used by the original construction heuristic `onestepCD` .

Average computation times for the `TS-PCP` heuristic are plotted in Figure 6. Computation times increase quadratically with the number of components. We notice that the average times to find the best solutions are much smaller than the average times to finish.

In the last experiment in this section, we randomly generated graphs with 1000 nodes distributed over 500 components with two nodes each. The edge density ranges from 0.1 to

	$F_{end} = 1$			$F_{end} = 5$			$F_{end} = 10$		
TabuTenure	min	avg	max	min	avg	max	min	avg	max
$U[\frac{1}{4}C', \frac{3}{4}C']$	53	53.5	54	52	52.7	53	52	52.5	53
$U[0, C']$	53	53.7	55	52	52.9	53	52	52.3	53
$U[0, \frac{1}{2}C']$	52	<u>53.1</u>	54	51	<u>52.2</u>	53	51	<u>51.3</u>	52
$U[\frac{1}{2}C', C']$	54	54.2	55	53	53.3	54	53	53.0	53
$U[\frac{1}{4}C', C']$	53	53.8	54	52	53.0	54	52	52.8	53
$U[0, \frac{3}{4}C']$	52	53.3	54	51	52.5	53	51	52.2	53

Table 2: Results for instance dsjc500.5-1, with $C' = \texttt{max}C - 1$.

	$F_{end} = 1$			$F_{end} = 5$			$F_{end} = 10$		
TabuTenure	min	avg	max	min	avg	max	min	avg	max
$U[\frac{1}{4}C', \frac{3}{4}C']$	47	47.8	49	46	46.8	47	46	46.7	47
$U[0, C']$	46	47.5	48	46	46.8	47	46	46.7	47
$U[0, \frac{1}{2}C']$	47	<u>47.3</u>	48	45	<u>46.1</u>	47	45	<u>45.9</u>	46
$U[\frac{1}{2}C', C']$	48	48.1	49	47	47.7	48	47	47.3	48
$U[\frac{1}{4}C', C']$	47	47.9	49	47	47.3	48	46	46.9	47
$U[0, \frac{3}{4}C']$	46	47.5	48	46	46.6	47	46	46.2	47

Table 3: Results for instance dsjc500.5-2, with $C' = \texttt{max}C - 1$.

	$F_{end} = 1$			$F_{end} = 5$			$F_{end} = 10$		
TabuTenure	min	avg	max	min	avg	max	min	avg	max
$U[\frac{1}{4}C', \frac{3}{4}C']$	44	44.8	45	44	44.4	45	44	44.0	44
$U[0, C']$	45	45.4	46	44	44.7	45	44	44.2	45
$U[0, \frac{1}{2}C']$	44	<u>44.6</u>	45	43	<u>43.7</u>	44	43	<u>43.2</u>	44
$U[\frac{1}{2}C', C']$	45	45.8	46	44	44.9	45	44	44.8	45
$U[\frac{1}{4}C', C']$	44	45.5	46	44	44.7	45	44	44.2	45
$U[0, \frac{3}{4}C']$	44	44.8	45	43	44.0	45	43	43.9	44

Table 4: Results for instance dsjc500.5-3, with $C' = \texttt{max}C - 1$.

	$F_{end} = 1$			$F_{end} = 5$			$F_{end} = 10$		
TabuTenure	min	avg	max	min	avg	max	min	avg	max
$U[\frac{1}{4}C', \frac{3}{4}C']$	43	43.5	44	42	42.8	43	42	42.4	43
$U[0, C']$	43	43.6	44	42	42.7	43	42	42.7	43
$U[0, \frac{1}{2}C']$	42	<u>42.8</u>	43	42	<u>42.0</u>	42	42	<u>42.0</u>	42
$U[\frac{1}{2}C', C']$	43	43.9	44	43	43.0	43	43	43.0	43
$U[\frac{1}{4}C', C']$	43	43.6	44	42	42.9	43	42	42.8	43
$U[0, \frac{3}{4}C']$	42	43.0	44	42	42.4	43	42	42.2	43

Table 5: Results for instance dsjc500.5-4, with $C' = \texttt{max}C - 1$.

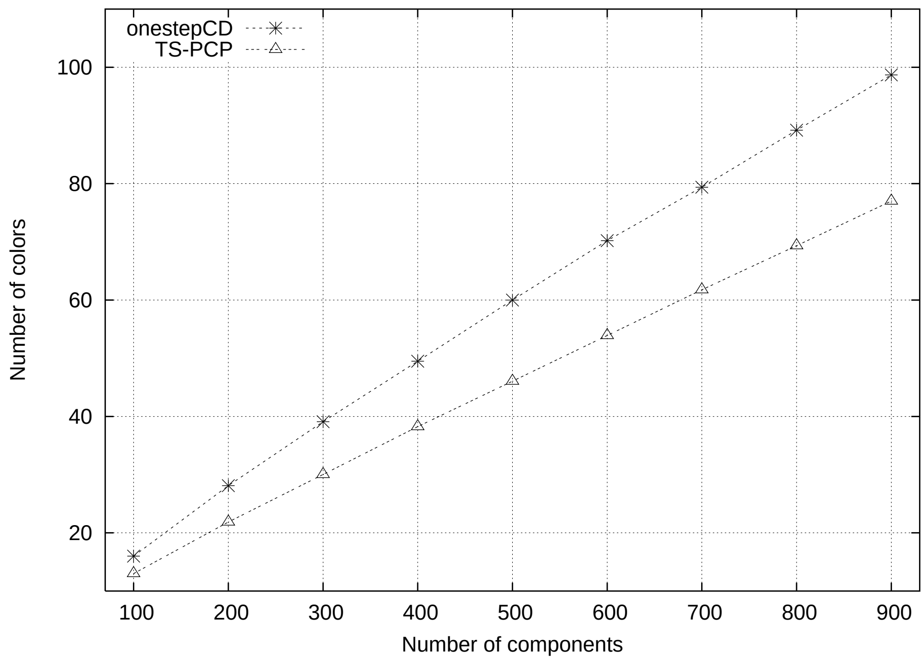

Figure 5: Average number of colors vs. number of components.

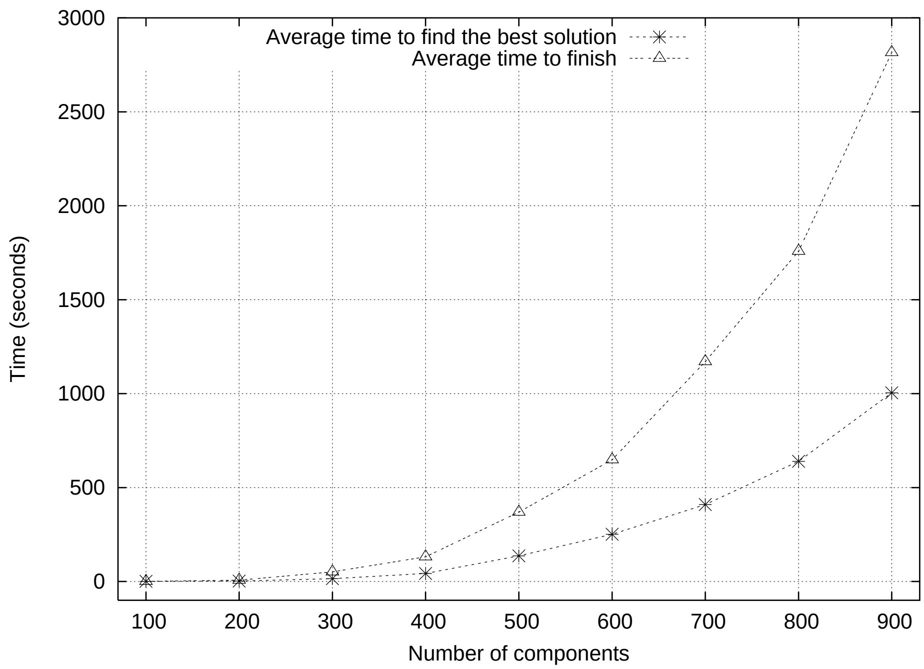

Figure 6: Average computation times for `TS-PCP` vs. number of components.

0.9. Ten graphs were generated for each density. Algorithms `onestepCD` and `TS-PCP` were run 10 times for each instance with different seeds. We observe in Figure 7 that the average number of colors increases with edge density. Figure 8 shows that once again the computation times increase quadratically with the edge density.

4.2 Application to RWA

In this section we report computational results concerning the proposed approach of combining the heuristics k-`EDR` with $k = 2$ and `TS-PCP` to solve problem min-RWA of routing and wavelength assignment in all optical networks.

We considered five realistic networks with arbitrary topologies, whose main characteristics are summarized in Table 6. All links are bidirectional and there are connections to be established between all pairs of nodes. Lightpaths are directional, i.e. the lightpath from node $i \in X$ to node $j \in X$ may be different from that connecting j to i. Accordingly, there are $|X| \cdot (|X| - 1)$ lightpaths to be established. Lower bounds were obtained as the solution of the linear relaxation of an integer programming formulation of min-RWA.

Name	Nodes	Links	Connections	Lower bound
NSFnet	14	21	182	13
USA	32	50	992	69
Brazil	27	70	702	24
Finland	31	51	930	46

Table 6: Characteristics of min-RWA instances.

The two approaches were applied to the four instances described in Table 6. Two hundred independent runs for each approach were performed for each instance. Execution was terminated when a solution of value less than or equal to some target value was found. We used target values of 13 for instance NSFnet, 70 for instance USA, 24 for instance Brazil, and 50 for instance Finland. These targets are sub-optimal values obtained as follows. Each approach was applied 20 times to each instance, with an upper bound of 10 minutes on the computation time. The selected target for each instance is the best solution value found by both heuristics in all instances. Empirical probability distributions for the time-to-target-solution-value are plotted. To plot the empirical distribution for each approach and each instance, we follow the methodology described in [1] and successfully applied to several other problems, see e.g. [7, 13]. We associate with the i-th smallest running time t_i a probability $p_i = (i - \frac{1}{2})/200$, and we plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, 200$. The results obtained by the combined approach `2-EDR` + `TS-PCP` proposed in this work are compared with those obtained by the multistart heuristic `Greedy-EDP-RWA` proposed in [12].

Figures 9 to 12 illustrate the topology of each network and display the time-to-target-solution-value plot for each approach. The new combined strategy `2-EDR` + `TS-PCP` is much more robust than the multistart heuristic `Greedy-EDP-RWA`. The former is more robust and finds target solutions much faster than the latter. `2-EDR` + `TS-PCP` clearly outperforms `Greedy-EDP-RWA` for instances Brazil and NSFnet. In the case of instances USA and Finland, the approach `2-EDR` + `TS-PCP` is much more stable than the multistart heuristic, whose

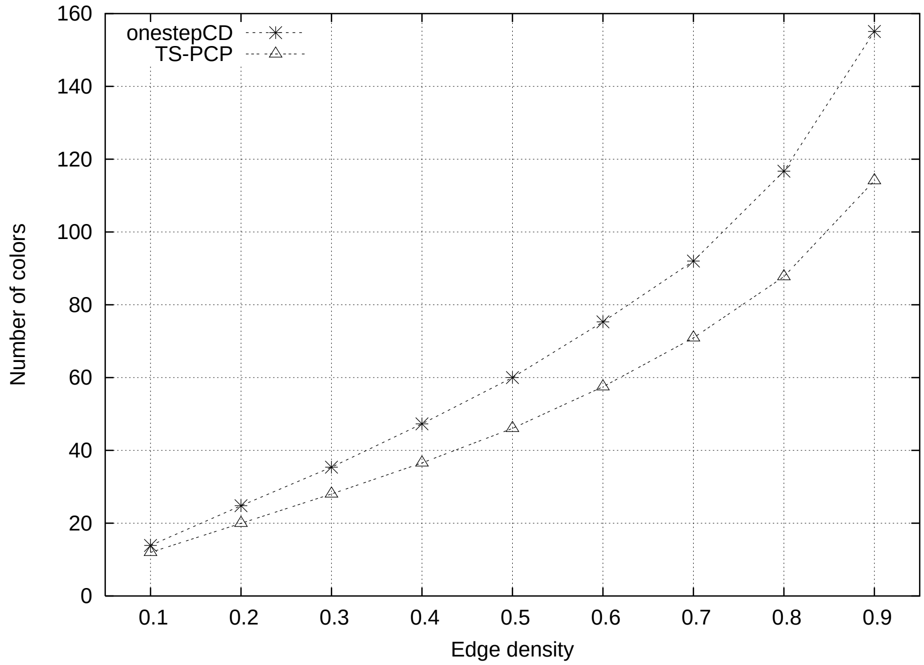

Figure 7: Average number of colors vs. edge density.

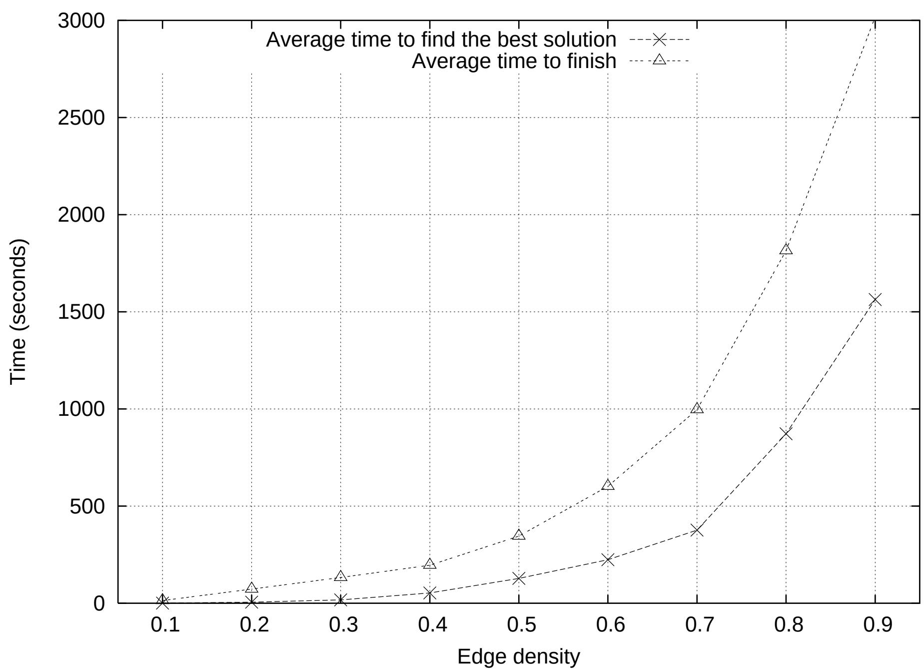

Figure 8: Average computation times for `TS-PCP` vs. edge density.

times to target solution values are much more spread and unpredictable.

5 Concluding Remarks

References

[1] R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. Probability distribution of solution time in GRASP: An experimental investigation. *Journal of Heuristics*, 8:343–373, 2002.

[2] C.R. Aragon, D.S. Johnson, A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; Part II, Graph coloring and number partitioning. *Operations Research*, 39:378–406, 1991.

[3] D. Bannerjee and B. Mukherjee. Practical approach for routing and wavelength assignment in large wavelength routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14:903–908, 1995.

[4] D. Brelaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22:251–256, 1979.

[5] J.S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su. A functional classification of routing and wavelength assignment schemes in DWDM networks: Static case. In *Proceedings of the 7th International Conference on Optical Communication and Networks*, pages 1109–1115, 2000.

[6] T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255:33–50, 2001.

[7] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the MAX-CUT problem. *Optimization Methods and Software*, 7:1033–1058, 2002.

[8] F. Glover and M. Laguna. *Tabu Search*. Kluwer, 1997.

[9] E. Hyytiã and J. Virtamo. Wavelength assignment and routing in WDM networks. In *Nordic Teletraffic Seminar 14*, pages 31–40, 1998.

[10] J.M. Kleinberg. *Approximation algorithms for disjoint paths problems*. PhD thesis, MIT, Cambridge, 1996.

[11] G. Li and R. Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Proceedings of the First Workshop on Optical Networks*, pages ??–??, 2000.

[12] P. Manohar, D. Manjunath, and R.K. Shevgaonkar. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *IEEE Communications Letters*, 5:211–213, 2002.

[13] M.G.C. Resende and C.C. Ribeiro. A GRASP with path-relinking for private virtual circuit routing. *Networks*, 41:104–114, 2003.

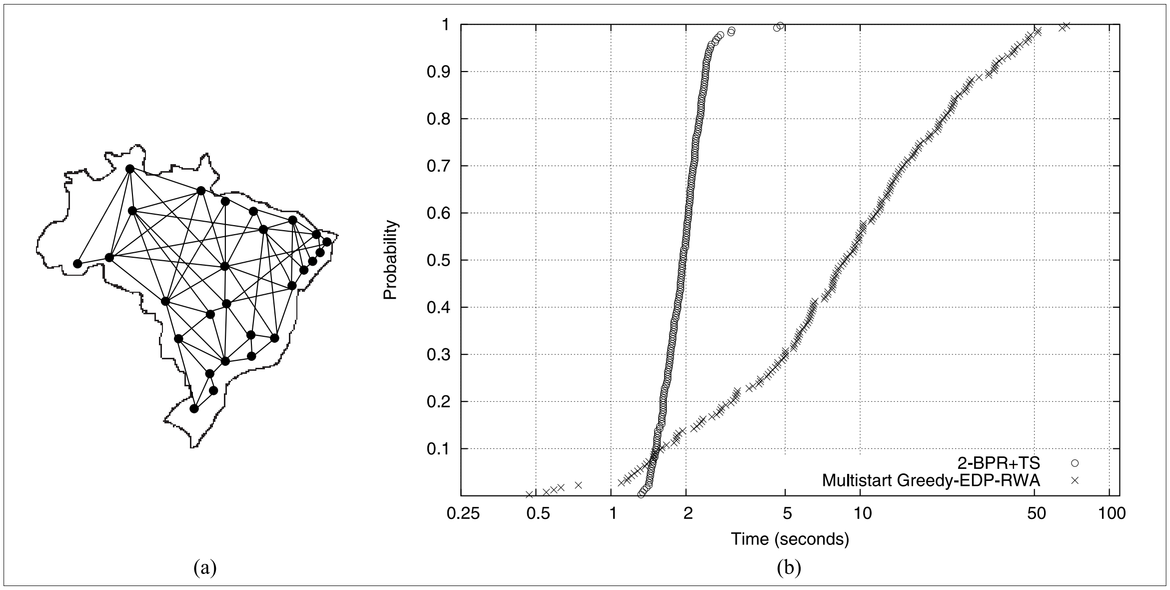

Figure 9: Instance Brazil.

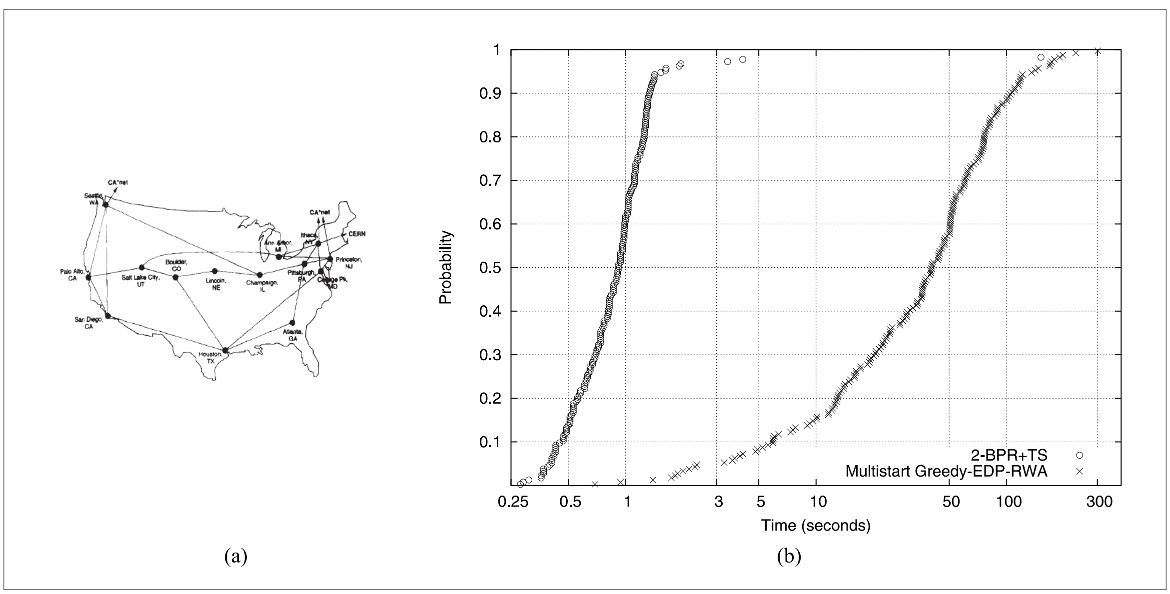

Figure 10: Instance NSFnet.

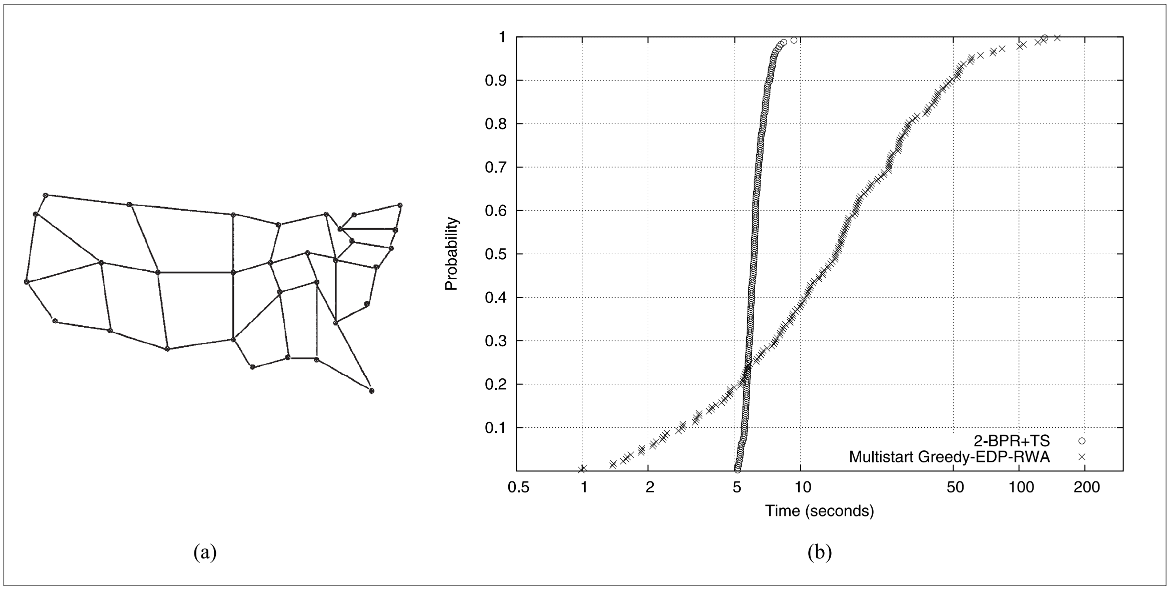

Figure 11: Instance USA.

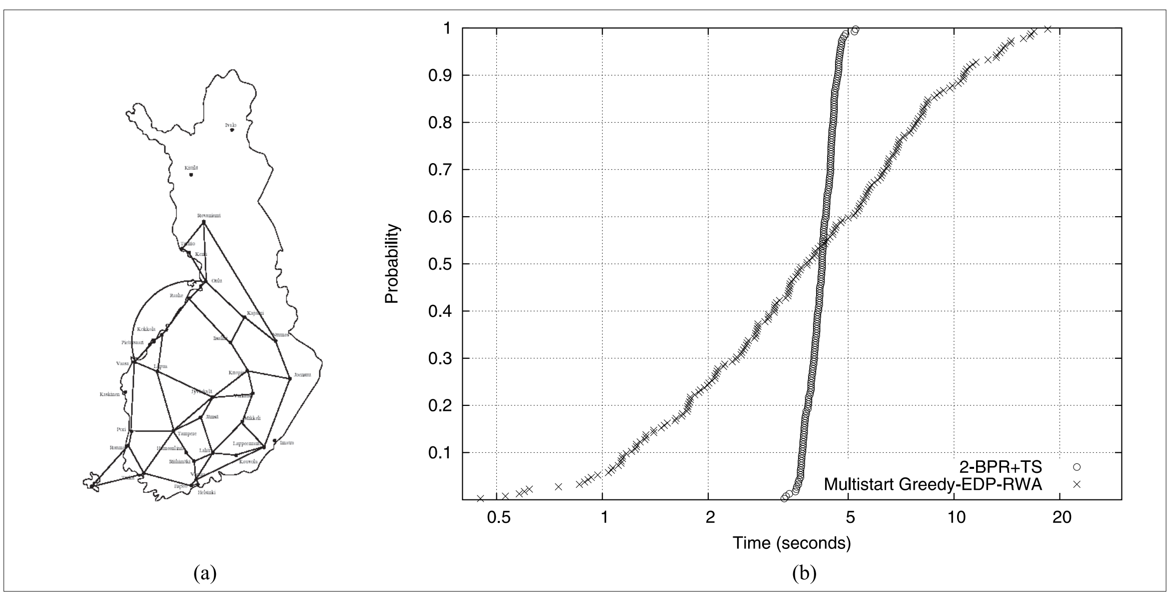

Figure 12: Instance Finland.

[14] J. Yoo and S. Banerjee. Design, analysis, and implementation of wavelength-routed all-optical networks: Routing and wavelength assignment approach. *IEEE Communications Surveys Broadband Networks Area*, ??:??–??, 1997.

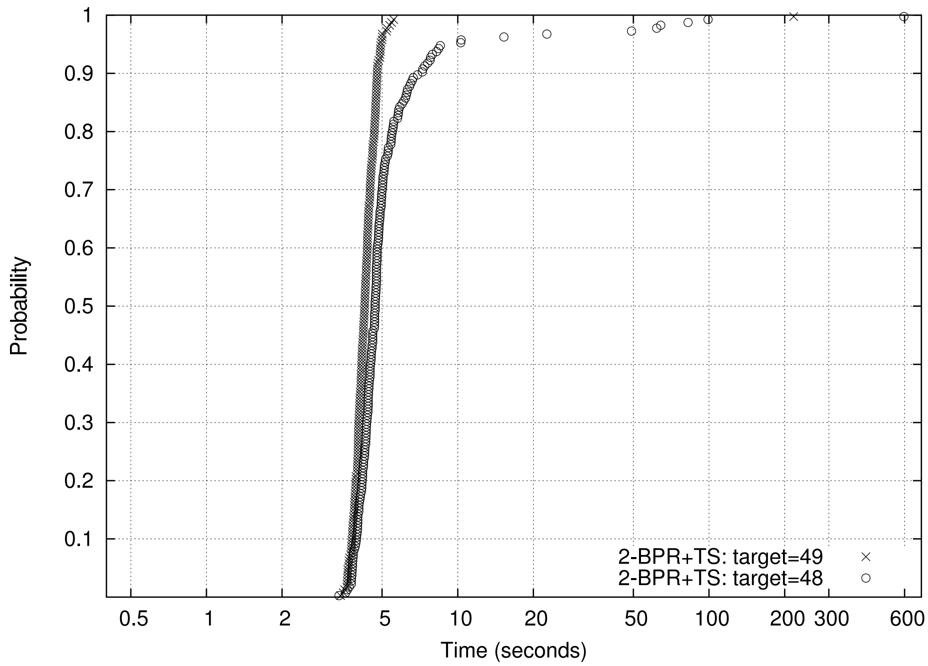

Figure 13: Rede Fin2