2014 10th International Conference on Natural Computation

GRASP for WDM Network Design Problem With Traffic Grooming

Xinyun Wu, Zhipeng lü, Qi Guo, Tao Ye
SMART, School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, P.R.China

***Abstract*—Two important problems arise in WDM network planning: network design to minimize the construction cost and traffic grooming to maximize the usage of high capacity channels. They are usually settled simultaneously and are denoted as the network design problem with traffic grooming (NDG). This paper proposes an approach based on Greedy Randomized Adaptive Search Procedure (GRASP) to solve the NDG problem. A mathematical formulation of the NDG problem is presented and a set of benchmarks generated according to real application scenarios are given for future research in this area, which are missing in the previous literature. Computational results show the effectiveness of the proposed GRASP algorithm by comparison with the public software CPLEX and the lower bound. This paper also shows an analysis on explaining the essential ingredients of the algorithm.**

***Index Terms*—WDM Mesh Network, traffic grooming, GRASP, heuristic, mathematical formulation**

I. Introduction

The network design problem arises in telecommunication where the construction cost should be controlled within a reasonable limit. A wavelength division multiplexing(WDM) network involves three layers, which are the traffic demands between each nodes, the optical network presenting the light-paths carrying the traffics between each pair of nodes, and the physical network carrying traffics in the WDM network. The physical network seldom changes when it has been constructed, while the optical network varies according to the different traffic demands.

The edges in the optical network called lightpaths are generated by two transceivers at the source and the sink nodes transferring traffics. Generally, the lightpath have a immense capacity compared to the bandwidths of the traffics. A technique referred as traffic grooming is adopted by telecommunication carriers in order to maximize the use of the lightpaths. Thus, a lightpath can be shared by several traffics.

In real applications, the cost of the transceivers is a dominant cost in the WDM network. Since each lightpath corresponds to two transceivers, the number of transceivers is twice the number of lightpaths. Therefore, providing each traffic an exclusive lightpath is rather expensive.

An optimization appears that given the physical network and the traffic demands the goal is to design an economical optical network which is able to carry all the traffic demands on the physical network.

There are numbers of studies related to this issue. Wang and Gu proved the NP-completeness of the traffic grooming problem in [1], implying that effective optimization algorithms are highly demanded for this challenging problem. Chen and Rouskas presented an effective and efficient hierarchical traffic grooming framework for WDM networks [2]. Saleh and Kamal addressed the problem of designing and provisioning of WDM networks to support many-to-many traffic grooming aiming at minimizing the overall network cost [3]. They also introduced two novel approximation algorithms for the many-to-many traffic grooming problem [4]. A mathematical formulation of the traffic grooming and routing problem was presented in [5] by Zhu and Mukherjee, and several fast heuristics were proposed and evaluated. Hu formulated the Grooming routing and wavelength assignment problem as Integer Linear Programming and divided it into two sub problems: the traffic grooming, routing problem and the wavelength assignment problem [6]. In [7], Thiagarajan and Somani proposed a connection admission control scheme to ensure fairness in terms of connection blocking. Srinivasan and Somani presented a theoretical capacity correlation model to calculate the blocking probability for WDM networks with constrained grooming capability [8]. In [9], Rubio-Largo and Vega-Rodriguez solved the traffic grooming problem by proposing two novel multi-objective evolutionary algorithms, showing the excellent properties of the proposed meta-heuristic algorithms.

In spite of the numerous studies on the WDM networks and the traffic grooming problem, there is no unified mathematical model for the problem. And there is lack of comparison platforms to evaluate the algorithms, while there is no efficient exact algorithm specified for the problem. In this paper, we study the WDM network design problem with traffic grooming (NDG), which aims to design a network (the optical layer of the WDM network) with the minimum number of lightpaths while satisfying the grooming constrains. The NDG problem is quit similar to the Multi-commodity Capacitated Network Design problem(MCND)[10], but there are some differences. Firstly, The network considered in NDG problem may be a complete graph, indicating that there may be much more design variables than the MCND problem. Moreover, The flow and the design variables are both integers in the NDG problem, while the MCND problem is different, with the former continuous and the latter discrete. These characteristics make the NDG problem particularly difficult.

We present an integer program for the NDG problem. The model provides a representation for the optical network and

(a) Solution I (b) Solution II

Fig. 1: An example for the NDG

the paths for all the traffics. In order to solve the NDG problem we adopt GRASP (Greedy Randomized Adaptive Search Procedures) [11] which is a meta-heuristic implemented in a wide range of applications. To test the performance, the proposed GRASP algorithm was applied to tackle several benchmark instances showing its high efficiency by comparing to the public software CPLEX and the lower bound. These instances are generated according to the real application scenarios, and can be used for the comparison with other algorithms in future research.

II. The NDG Problem

A. Problem description

The networks design problem with traffic grooming is defined as follows. Given node set $V = \{1, 2, ..., n\}$ and traffic set T. The objective is to design a network, which is denoted as an undirected graph $G = (V, E)$, with the minimum edges according to the node set V and the traffic set T and make all the traffics groomed on the designed network. The traffics represent point-to-point traffics such that a traffic $t = (s_t, d_t, c_t)$ with bandwidth c_t is routed through a single path from source node s_t to sink node d_t. The edges in E are denoted as lightpaths. All lightpaths carry traffics no more than the capacity limit C.

For example, given a node set V with 4 nodes and a traffic set T with 5 traffics of bandwidth 1 and assuming that the capacity of each lightpath is 4, the straightforward solution of this problem is to construct a network with 5 lightpaths, each carrying one traffic, as shown in Fig. 1a. The grey bold solid lines represent the lightpaths connecting each nodes. The color lines represent the routes for each traffic. As an alternative, we can groom traffic $(A, C, 1)$ to lightpath AD and DC and groom traffic $(B, C, 1)$ to lightpath AB, AD and DC, which means the traffic $(A, C, 1)$ routes $A-D-C$ and traffic $(B, C, 1)$ routes $B-A-D-C$. Thus only 3 lightpaths are enough to satisfy all the 5 traffics, saving 2 lightpaths than the previous solution. Obviously, the network with 3 lightpaths is the optimal solution for this case.

B. Problem formulation

The WDM network design problem consists of designing a WDM network with the minimum number of lightpaths and grooming all the traffics in set T. A candidate solution is represented by two matrices: $n \times n$ matricx $[x]$ where $x_{ij} = 1$ only if there exists a lightpath connecting nodes i and j. $m \times n \times n$ matrix $[y]$ where $y_{tij} = 1$ only if traffic t traverses lightpath connecting nodes i and j. The graph represented by $[x]$ is an undirected graph. However, it is noteworthy that the graph denoted by $[y]$ is a digraph, which represent the path of all traffics on the WDM network.

TABLE I: Notations used for the NDG problem

Symbols	Description
n	The number of nodes
m	The number of traffics.
C	The capacity for each lightpath
T	Set of traffics, $t \in T, t = (s_t, d_t, c_t)$
s_t	The source node of traffic t
d_t	The sink node of traffic t
c_t	The bandwidth of traffic t
b_{tj}	Mark the source node and the sink node of a traffic $b_{tj} = \begin{cases} 1 & \text{, for } j = d_t \\ -1 & \text{, for } j = s_t \\ 0 & \text{, otherwise} \end{cases}$
$[x]_{n\times n}$	Decision matrix for the lightpaths
$[y]_{m\times n\times n}$	Decision matrix for the lighpath route of the traffics

The symbol and variable definitions are presented in TABLE I. Given these notations, we can describe the WDM network design problem in a formal way as follows:

minimize: $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij}$

subject to:

$$x_{ij} = x_{ji} \quad i, j = 1, 2...n \tag{1}$$

$$y_{tij} - x_{ij} \le 0 \quad t \in T \quad i, j = 1, 2...n \tag{2}$$

$$\sum_{i=1}^{n} y_{tij} - \sum_{i=1}^{n} y_{tji} = b_{tj} \quad \forall t \in T \quad j = 1, 2...n \tag{3}$$

$$\sum_{t\in T} c_t (y_{tij} + y_{tji}) \le C \quad i = 1, 2, ..., n-1 \; j = i+1..n \tag{4}$$

$$x_{ij}, y_{tij} \in \{0, 1\} \quad t \in T \quad i, j = 1, 2...n \tag{5}$$

Constraint (1) guarantees that the graph that $[x]$ represents is undirected. Constraint (2) ensures that the lightpath carrying a traffic is an available one. Constraint (3) require that the route of a traffic is an available one. Constraint (4) ensures that each lightpath can only carry traffics not more than its capacity.

III. Solution method

In order to better describe our proposed Greedy Randomized Adaptive Search Procedure(GRASP) for the WDM network design problem, we first define some variables as shown in TABLE II.

The presented GRASP algorithm follows a general framework which is an iterative process where each iteration consists of two phases, a construction phase (Section III-A) and a local search phase(Section III-B). The best solution found so far is kept as the result. The general framework of our GRASP algorithm is described in Algorithm 1. The GRASP iterations take place in lines 3-17, and terminate when some termination

TABLE II: Variable used in the GRASP algorithm

Symbols	Description
T_u	The set consists of the traffics which have not been groomed.
O_{ij}	The overload amount of the lightpath connecting nodes i and j.
f_g	The objective for procedure *Grooming*.
Δf_g	The change of f_g.
L_u	The set consists of the indexes of lightpaths.
l	Index of a lightpath.
c_l	The free bandwidth of lightpath l.
t	Index of a traffic.
P	A series of lightpaths that can represent a path for a traffic.
$[x]_b$	A backup for $[x]$.
$[y]_b$	A backup for $[y]$.
mv	A move can lead to a new solution denoted by $S \oplus mv$ by applying it to the solution S
$N_{LM}([x])$	The neighborhood of the solution in *Lightpaths-Minimization*.
$N_g([y])$	The neighborhood of the solution in *Grooming*.
0	Zero matrix.

Algorithm 1 The Main Framework

```
1: procedure MAIN(V, T)
2:     bestObj ← ∞
3:     repeat
4:         [x], [y] ← CONSTRUCTION([x], [y], T)
5:         [x], [y] ← LIGHTPATHMIN([x], [y], T)
6:         currObj ← COUNTLIGHTPATHS([x])
7:         if currObj ≤ bestObj then
8:             [x]_b, [y]_b ← [x], [y]
9:             bestObj ← currObj
10:        else
11:            [x], [y] ← [x]_b, [y]_b
12:        end if
13:        if termination condition met then
14:            EXPORTSOLUTION([x], [y])
15:            EXIT
16:        end if
17:    until Forever
18: end procedure
```

Algorithm 2 Construction

```
procedure CONSTRUCTION([x], [y], T)
    [y] ← 0
    SHUFFLETRAFFICS(T)
    for all t ∈ T do
        if ∃P connecting s_t and d_t AND c_l ≥ c_t ∀l ∈ P then
            ASSIGN([y], t, P)
        else
            Flag ← false
        end if
        if Flag = flase then
            x_{s_t d_t} ← 1
            x_{d_t s_t} ← 1
            P ← {The new l connecting s_t and d_t}
            ASSIGN([y], t, P)
        end if
    end for
    return [x], [y]
end procedure
```

criterion, such as the time limit, is satisfied. Line 4 is the GRASP construction phase, while line 5 is the local search phase. The rest lines check whether an improved solution is found. In the following sections we present more detailed descriptions for these two phases.

Algorithm 3 Lightpath-Minimization

```
procedure LIGHTPATHMIN([x], [y], [N])
    L_u ← INDEXOFLIGHTPATHS([x])
    while L_u ≠ φ do
        [x]_b, [y]_b ← [x], [y]
        l ← CHOOSELIGHTPATH(L_u)
        L_u ← L_u − l
        T_u ← Traffics passing through l
        [x], [y] ← REMOVELIGHTPATH([x], [y], l)
        flag ← GROOMING([x], [y], T_u)
        if flag = TRUE then
            EXPORTSOLUTION([x], [y])
        else
            [x], [y] ← [x]_b, [y]_b
        end if
    end while
end procedure
```

A. Construction

The construction phase of our GRASP algorithm generates a feasible crude solution iteratively. At each construction iteration, one traffic is chosen and assigned a path on the network using a greedy heuristic and a new lightpath is introduced for the traffic if the assignment fails, which is shown in Algorithm 2. The traffics should be shuffled before the construction. Then each time the construction phase will construct a solution different from the previous iteration.

It is noteworthy that in the above procedure, we need to examine the other paths between two nodes if the shortest path do not meet the requirement. Thus an algorithm is needed to find all the simple paths between two nodes. In this paper, we implement a *K shortest paths algorithm* described in [12] for finding these candidate paths.

B. Lightpath-Minimization

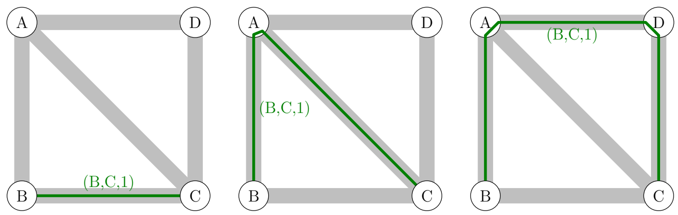
The *Lightpath-Minimization* phase is aimed at decreasing the number of lightpaths and is composed of two local search algorithms in a multi-level architecture. The high level local search algorithm is a topology transformation algorithm, while the low level local search algorithms (Section III-C) solve the grooming problem. The pseudo-code of the *Lightpath-Minimization* phase is shown in Algorithm 3.

The objective function (denoted as $currObj$) is defined as the number of lightpaths currently in use:

$$currObj = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij}$$

It is widely believed that one of the most important features of a local search algorithm is the definition of its neighborhood. In a local search procedure, applying a move mv to a solution S leads to a new solution denoted by $S \oplus mv$. Let $M(S)$ be the set of all possible moves which can be applied to S, then the neighborhood N of S is defined by: $N(S) = \{S \oplus mv | mv \in M(S)\}$.

In our GRASP algorithm, there are two neighborhood structures respectively for the two local search algorithms. The neighborhood structure $N_{LM}([x])$ used in *Lightpaths-Minimization* for the high level local search is composed of all possible moves of *lightpath-deletion*, where *lightpath-deletion*



(a) origin (b) change I (c) change II

Fig. 2: *RouteChange* illustration

is defined as removing one lightpath from the current virtual topology. Therefore, the size of the neighborhood $N_{LM}([x])$ is bounded by $O(\mu)$, where μ is the number of lightpaths.

More specifically, let L_u denote the indexes of all the lightpaths. At each iteration, we choose one lightpath from L_u and tentatively remove the lightpath from the current solution. Then, we try to reassign those traffics carried by the removed lightpath on the network by calling the subroutines *Grooming*. Subroutine *Grooming* assigns these traffics to the network, ensuring that the capacity constraint is satisfied (constraint (4)). An improved solution is gained if subroutine *Grooming* returns TRUE.

In this high level local search we accept a candidate solution in the current neighborhood immediately if the deletion of one lightpath l can satisfy all the constraints, which means we use a first-improvement strategy based simple descent algorithm. Then, L_u is updated by removing lightpath l. The above procedure is repeated until it fails for the deletion of any lightpath in L_u.

C. Grooming

As a subroutine of *Lightpath-Minimization*, the *Grooming* procedure can be considered as a low level local search algorithm in our GRASP algorithm. Once the topology changes by removing one lightpath, the *Grooming* procedure is called to verify if the remained network can still satisfy the capacity constraint. The main purpose of *Grooming* is to groom traffics to lightpaths such that all the lightpaths are not overloaded.

In the *Grooming* procedure, the objective function (denoted as f_g) is defined as the violation of the capacity constraints for all the lightpaths:

$$f_g = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} O_{ij}$$

where O_{ij} represents the overload amount of the lightpath connecting nodes i and j.

$$O_{ij} = \max\left(\sum_{t \in T}(c_t \cdot y_{tij}) + \sum_{t \in T}(c_t \cdot y_{tji}) - C, 0\right)$$

One observes that $f_g = 0$ corresponds to a network which satisfies all the capacity constraints.

The neighborhood used in the *Grooming* procedure, which is denoted as $N_g([y])$, is composed of all feasible moves of *RouteChange*, where a *RouteChange* move changes the route of one traffic to another candidate route of this traffic. Applying a *RouteChange* move to traffic t in the solution consists in modifying the values of y_{tij} $(1 \le i \le n, 1 \le j \le n)$. Therefore, the size of neighborhood $N_g([y])$ is bounded by $O(m \cdot k)$ where $m = |T|$ and k is the maximum number of candidate routes for one traffic.

For example, Fig.2 depicts how *RouteChange* works for traffic $t = (B, C, 1)$. There are 3 candidate routes for traffic t, which are B-C, B-A-C and B-A-D-C. In the current solution, the route for traffic t is B-C (illustrated by green line in Fig.2a). The *RouteChange* mv_g may change the route of t to another candidate route of t which is either B-A-C (Fig.2b) or B-A-D-C (Fig.2c). The *K shortest path algorithm* is also needed here to find the candidate paths for each traffic.

In *Grooming* procedure, we first assign the traffics in T_u with random paths. Then we use a best-improvement based simple descent algorithm to try to make the solution feasible. That is to say we choose the candidate solution in $N_g([y])$ with the minimum Δf_g value. The procedure stops when $f_g = 0$ or the minimum Δf_g value is non-negative.

IV. Experimental results

In this section, we report intensive experimental results of the proposed GRASP algorithm on two sets of totally 6 instances and compare our results with the results obtained by the commercial software CPLEX or a rough estimated lower bound[1].

A. Experimental protocol

The presented GRASP algorithm is programmed in C++ and compiled using GNU GCC on a PC running Windows 7 with 3.1GHz CPU and 6.0Gb RAM. Two sets of test problem instances are considered in our experiments, in total constituting 6 instances. The first set of benchmarks is composed of 2 small instances of size $n = 8$ and $m = 15-50$. The second set of benchmarks consists of 4 large problem instances which are generated according to the real application scenarios with size $n = 20$ and $m = 200$. To obtain our computational results, each instance is solved 20 times independently with different random seeds. Each run is stopped when the processing time reaches its time limit.

B. The lower bound

The public software IBM ILOG CPLEX optimization studio 12.4 and a rough estimated lower bound are used to assess the accuracy and efficiency of our GRASP algorithm.

For the first two instances with small problem scale, CPLEX is sufficient to get a solution. Since CPLEX can get the optimal solution, we compared the solution given by CPLEX with the solution obtained by our GRASP algorithm. The result shows the effectiveness of our GRASP algorithm.

For the last four instances, the problem size is too large to solve for CPLEX. CPLEX failed to get a solution good

[1]The tested instances can be found online(https://bitbucket.org/HUST-smartLab/groominginstances) or obtained from the author.

TABLE III: Result for the small-scale instances

instance	nodes	traffics	CPLEX	GRASP
s1	8	15	7	7
s2	8	20	9	9

TABLE IV: Result for the large-scale instances

Instance	LB	Best	Average	Init
G20_200_1	19	24(26%)	25	29
G20_200_2	19	24(10%)	26	31
G20_200_3	19	24(10%)	25	31
G20_200_4	19	22(5%)	25	32

enough for days and crashed due to lack of memory. So we use a rough estimated lower bound to evaluate the efficiency of our GRASP algorithm.

Assume n' nodes are involved in the traffic set T, then these n' nodes should be in the network of the final solution. And the minimum graph that contains n' nodes should have at least $n' - 1$ edges. So, $n' - 1$ can be seen as a lower bound for the NDG problem. This lower bound is a rather rough one, but it fits well for those instances which have few traffics.

C. *Result for the small-scale instances*

There are 8 nodes in both instances. The traffic set for each instance is generated by choosing pairs of nodes randomly. The lightpath capacity is set to be 4, and the bandwidth for each traffic is 1.

The results for this set of instances are shown in TABLE III. The column CPLEX represents the objective value given by CPLEX. The column GRASP represents the best objective value for each instance obtained by our GRASP algorithm.

The solutions for instances s1 and s2 are illustrated in Fig.3. In Fig.3, the gray bold lines represent the lightpaths, while the color curves represent the traffics. One observes that the traffics are well packed in these lightpaths, showing the high utilization of lightpaths. And our algorithm is able to obtain the objective value obtained by CPLEX, which means our algorithm is able to obtain the optimal solutions for these small instances.

D. *Results for the large-scale instances*

In this section, 4 large instances with 20 nodes and 200 traffics were used. In each instance, half of the traffics have a bandwidth of 1, and the remaining traffics have a bandwidth of 2. For all the instances, the lightpath capacity is set to be 32.

The computational results are shown in TABLE IV. Column **LB** gives the lower bounds for each instance. Column **Best** reports the best results obtained by our algorithm over 20 independent runs, the percentages in the brackets are the probability of getting this best objective value based on this experiment. Column **Init** presents the average objective value obtained at the first time execution of the construction phase and column **time** reports the time limit for the instance. All the average values are the smallest integral value that is greater than or equal to the specified decimal value.

From TABLE IV, one observes that our algorithm obtains good results which quit close to the lower bound. This experiment demonstrates that the refined local search phase is essential for our GRASP algorithm, while the enhanced construction strategy can also slightly improve our GRASP algorithm. Based on the above results, our GRASP algorithm is competitive not only for small scale instances but also for large scale instances.

V. Analysis and discussion

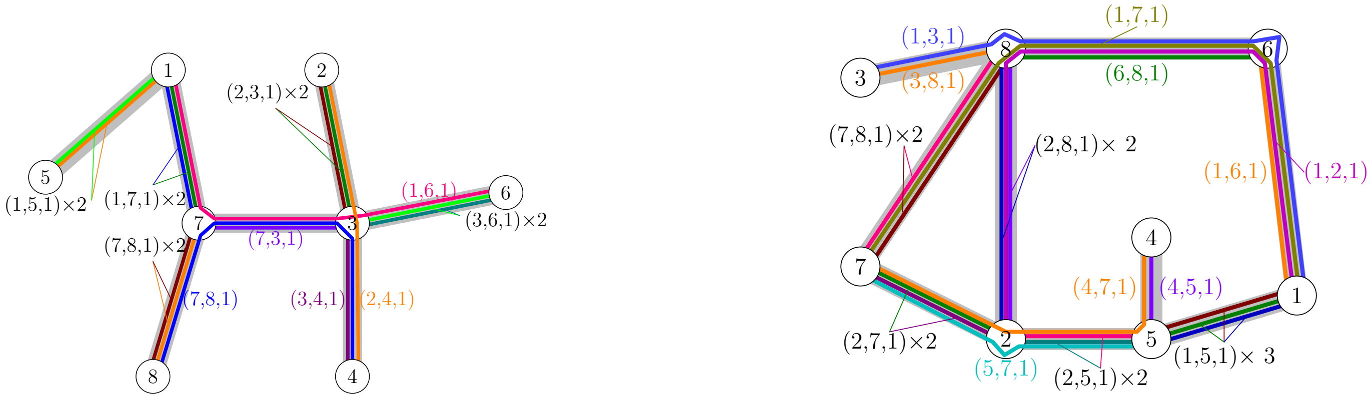
We turn now our attention to analyze some important features of the GRASP algorithm. In Section III-B, we introduced a multi-level local search procedure to minimize the number of lightpaths. Unlike the common local search procedure, *Lightpath-Minimization* consists of two sub local search procedures which are in a nested structure. The high level local search procedure is a topology transformation algorithm trying to delete the removable lightpaths, while the low level local search procedure try to verify the feasibility of this deletion. In the design of the algorithm, we found that the multi-level local search procedure gives a much clearer structure of the original problem and reduces the difficulty of designing the neighborhood structure which is a major technical problem in designing a local search algorithm.

From TABLE IV, we can see that *Lightpath-Minimization* improved the solution given by the construction phase significantly. This indicates that the *Lightpath-Minimization* has a high efficiency and constitutes an important part of our GRASP algorithm.

VI. Conclusion

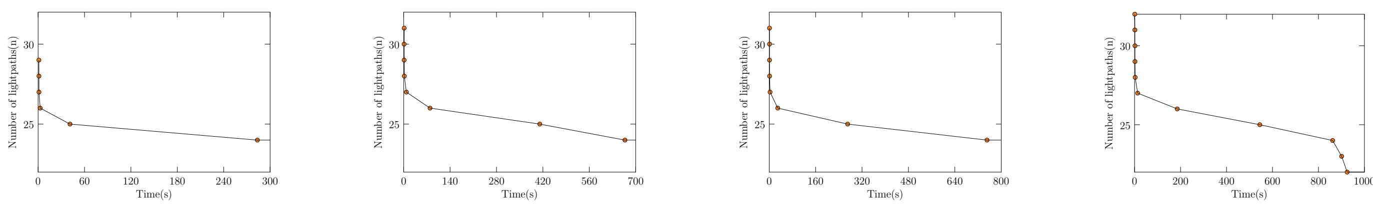
In this paper, we have presented the GRASP algorithm for tackling the WDM network design problem with traffic grooming. The NDG problem is a typical optimization problem which is a quite challenge topic with high complexity. It is widely considered in the optical communications industry as well as in academia. However, a unified model and the comparison platforms for the algorithms are absent in the previous literature.

By giving the mathematical formulation of the NDG problem, we present a GRASP algorithm for solving it. We also give some benchmark instances generated according to the real application scenarios to evaluate our proposed algorithm. The proposed GRASP algorithm follows a general framework which consists of two phases: construction and multi-level local search. Computational results show the effectiveness and efficiency of our GRASP algorithm for solving the tested instances by comparison with the CPLEX and the lower bound. This study reminds us that it is essential to introduce meaningful construction mechanisms and highlight the problem specific knowledge in designing meta-heuristic search algorithm. Following this spirit, we hope to design even more robust and effective meta-heuristic algorithms for solving the NDG problem as well as other optimization problems.



(a) s1 (b) s2

Fig. 3: The illustration for the solution



(a) G20_200_1 (b) G20_200_2 (c) G20_200_3 (d) G20_200_4

Fig. 4: Performance over time

REFERENCES

[1] Y. Wang and Q. Gu, "On the complexity and algorithm of grooming regular traffic in WDM optical networks," *Journal of Parallel and Distributed Computing*, vol. 68, pp. 877–886, 2008.

[2] B. Chen, G. Rouskas, and R. Dutta, "On hierarchical traffic grooming in wdm networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1226 –1238, oct. 2008.

[3] M. Saleh and A. Kamal, "Design and provisioning of wdm networks with many-to-many traffic grooming," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1869 –1882, dec. 2010.

[4] ——, "Approximation algorithms for many-to-many traffic grooming in optical wdm networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1527 –1540, oct. 2012.

[5] K. Zhu and B. Mukherjee, "Traffic grooming in an optical wdm mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 122 –133, jan 2002.

[6] J. Hu and B. Leida, "Traffic grooming, routing, and wavelength assignment in optical wdm mesh networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol. (xxxv+2866).

[7] S. Thiagarajan and A. K. Somani, "Capacity fairness of WDM networks with grooming capabilities," in *Storage and Retrieval for Image and Video Databases*, 2001.

[8] R. Srinivasan and A. Somani, "A generalized framework for analyzing time-space switched optical networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, 2001, pp. 179 –188 vol.1.

[9] A. Rubio Largo, M. Vega Rodriguez, J. Gomez Pulido, and J. Sanchez Perez, "Multiobjective metaheuristics for traffic grooming in optical networks," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, p. 1, 2012.

[10] B. Gendron, T. G. Crainic, and A. Frangioni, *Multicommodity capacitated network design*. Springer, 1999.

[11] T. Feo and M. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF01096763

[12] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. pp. 712–716, 1971. [Online]. Available: http://www.jstor.org/stable/2629312