Modèle d'optimisation pour la défragmentation de la capacité

Huy Duong, Brigitte Jaumard, David Coudert, Ron Armolavicius

Modèle d'optimisation pour la défragmentation de la capacité†

H. Duong[1] et B. Jaumard [1] et D. Coudert [2] et R. Armolavicius [3]

[1] *Computer Science & Software Engineering, Concordia University, Montréal, QC, Canada*
[2] *Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France*
[3] *Ciena, Ottawa, ON, Canada*

L'évolution du trafic peut entraîner une dégradation de l'utilisation des ressources d'un réseau liée à la fragmentation de la bande passante. En effet, la route d'une connexion (e.g., LSP dans un réseau MPLS) dépend des ressources disponibles lors de son établissement, et peut donc être éloignée du plus court chemin. La défragmentation de la capacité sur la couche utilisateur est alors nécessaire pour optimiser l'utilisation des ressources.

La reconfiguration vers un routage optimisé peut se faire à l'aide d'un processus itératif utilisant l'opération ≪ make-before-break ≫ (MBB). Le processus choisit une connexion, détermine une nouvelle route utilisant des ressources libres, bascule la connexion sur cette nouvelle route et libère les ressources qui ne sont plus utilisées. Puis, il considère une nouvelle connexion jusqu'à atteindre un routage optimisé. Le défi consiste à déterminer l'ordre dans lequel effectuer un nombre borné d'étapes de re-routages de connexions pour obtenir un routage optimisant l'utilisation des ressources.

Nous proposons un modèle d'optimisation exact de reconfiguration MBB minimisant l'utilisation des ressources. Nos résultats numériques montrent que nous améliorons l'état-de-l'art en résolvant des instances sur des réseaux à 30 nœuds.

Mots-clefs : Fragmentation des ressources; réseau optique; couche utilisateur; reroutage transparent; reroutage MBB

1 Introduction

Network reconfiguration is required in order to adapt to traffic changes, network failures, or deployment of new network resources. It occurs at the optical layer in order to make sure that the upper layer traffic, e.g., IP layer traffic, can be efficiently carried. In such a case, we deal with lightpath reconfigurations and the primary objective is to reduce disruptions to user traffic carried by existing lightpaths, measured by the number of disrupted lightpaths or the duration of lightpath disruptions [5]. Network reconfiguration may also appear in the logical layer, in order to attain a better resource utilization [3]. In heavily loaded networks, dynamic connection actions may result in a set of connections where some paths are not the shortest possible ones, leading to poor resource utilization compared to an optimized state. Thus, global connection re-optimization is proposed at certain time intervals to improve the network performance.

Researchers have investigated this connection re-optimization along two directions. The first one consists in computing an optimized provisioning with respect to resource utilization, and then finding an ordered sequence of connection rerouting in order to migrate from the current network provisioning to the optimized one with the minimum number of disruptions [1, 2, 6]. These studies usually have the constraint that a request can be rerouted at most once (i.e., from legacy to optimized route). As a result, it usually prevents the existence of a strategy using only make-before-break (MBB), i.e., the new route of connection k cannot be established using resources that are not used by any of the currently provisioned connections before to tear down the legacy route of k. Indeed, it is due to the presence of dependency cycles (e.g., request k_1 needs to be rerouted before k_2 because a link of the new route of k_2 belongs to the current route of k_1, and for similar reasons, k_2 needs to be rerouted before k_1). In order to find rerouting strategies, authors have proposed to use the break-before-make (BBM) paradigm that allows for the temporary interruption of requests, and so breaking dependency cycles [1, 2, 6].

†H. Duong was supported by a MITACS & Ciena Converge Fellowship. B. Jaumard has been supported by a Concordia Univ. Research Chair grant. D. Coudert was supported by ANR program "Investments for the Future" under ref. ANR-11-LABX-0031-01.

The second direction is to compute the best provisioning that is reachable from the legacy provisioning by a sequence of connection reroutings with no disruption, i.e., under the MBB paradigm. This NP-Complete optimization problem has received very little attention and the best known solution is the compact ILP formulation proposed by Klopfenstein [4]. In this paper, we propose a scalable optimization model for this problem (Section 2). Numerical results show that we improve very signifcantly against the state-of-the-art [4], enabling to solve instances on networks with up to 30 nodes (Section 3).

2 Make-Before-Break defragmentation model

We propose a decomposition model, called DEFRAG_RC, based on a set of rerouting configurations, where each rerouting configuration proposes a potential MBB rerouting of a single connection, i.e., a connection for which there exists an alternate route with enough spare bandwidth for its provisioning. The model is parameterized by $|T|$, a bound on the number of reroutings. The objective is to minimize the bandwidth requirements of the provisioning at time $|T|$. Observe that there is no guarantee to reach the best provisioning with a monotonous sequence, i.e., such that the overall bandwidth requirement decreases after each single connection rerouting. It may happen that the overall bandwidth requirement increases after a given connection rerouting in order to reach the minimum bandwidth provisioning at a later time, or at time $|T|$.

Notations. We consider a network represented by a directed multi-graph $G = (V,L)$ where V is the set of nodes (indexed by v) and L denotes the set of links (indexed by ℓ). Different links may exist between two nodes in order to model different logical links, with, e.g., different types of traffic. Let C_ℓ denote the transport capacity of link ℓ. Let K be the set of connection requests. Request $k \in K$ is characterized by its source (s_k), destination (d_k), and bandwidth requirement (b_k). We assume that the network undergoes a series of connection re-optimization at different time stamps (rerouting events). Let T (indexed by t) be the set of those time stamps, with $t = 0$ being the initial one. Let P be the overall set of potential rerouting operations, with $P = \bigcup_{k\in K} \bigcup_{t\in T} P_k^t$, where P_k^t is the set of potential routes of connection request k at time stamp t.

Modeling of DEFRAG_RC. The integer linear programming (ILP) formulation of DEFRAG_RC is the master problem (MP) in Figure 1. It uses the following binary variables :

- $z_{kp}^t = 1$ if route $p \in P_k$ is selected at time t for the rerouting of $k \in K$, 0 otherwise.
- C_ℓ^t = required bandwidth on link ℓ at time period t.

It also uses the following parameters :

- $a_{k\ell}^0 = 1$ if link $\ell \in L$ is used in the initial routing of connection request $k \in K$, 0 otherwise.
- $C_\ell^0 = \sum_{k\in K} b_k a_{k\ell}^0$ = initial bandwidth usage on link $\ell \in L$.
- $\delta_\ell^p = 1$ if path $p \in P$ uses link $\ell \in L$, 0 otherwise.

Our goal is to minimize the bandwidth usage (Objective (1)). Constraints (2) prevent from selecting more than one rerouting operation at each time period. $|T| \leq |K|$ is an upper bound on the number of rerouting operations as we cannot predict the number of required MBB reroutings. Constraints (3) ensure that a connection request is rerouted at most once. Constraints (4) make sure that transport capacities are never exceeded. Constraints (5) update the bandwidth usage on link ℓ at time period t, taking into account the unique connection request that has been rerouted at t. Constraints (6)-(7) define the domain of the variables.

Column generation. The model (1)-(7) has an exponential number of variables, and therefore column generation is required. We decompose the original problem into a *restricted master problem* (RMP), i.e., model (1)-(7) with a very restricted number of variables, and a *pricing problem* (PP). The RMP and the PP are solved alternately. Solving the RMP consists in selecting the best connection reroutings, while solving the PPs allows for the generation of improving potential reroutings The process continues until the optimality condition is satisfied. An ε-optimal solution is derived by solving exactly the ILP model associated with the last RMP, with $\varepsilon = (\tilde{z}_{\text{ILP}} - z^\star_{\text{LP}})/z^\star_{\text{LP}}$, where $z^\star_{\text{LP}}$ and $\tilde{z}_{\text{ILP}}$ denote the optimal LP value and the optimal ILP value of the last RMP, respectively. The solution process is illustrated in the flowchart of Figure 1.

Pricing Problem, PP$_t$. Let $u_t^{(2)} \leq 0$, $u_k^{(3)} \leq 0$ and $u_{\ell t}^{(5)} \gtreqless 0$ be the values of the dual variables associated with Constraints (2), (3) and (5), respectively. We use the following binary variables :

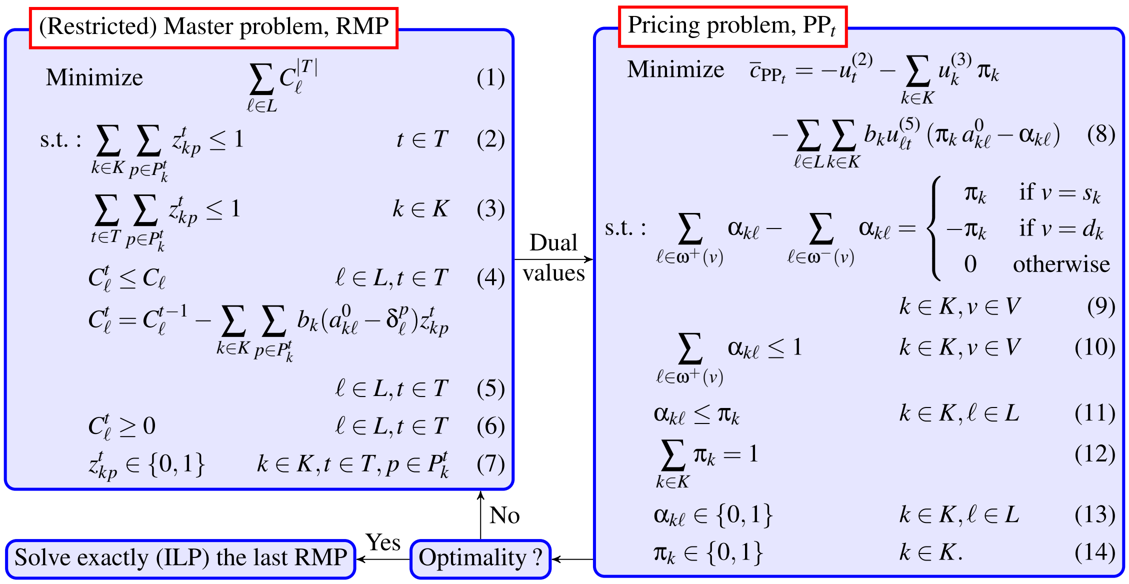


FIGURE 1: Flow chart of decomposition model DEFRAG_RC

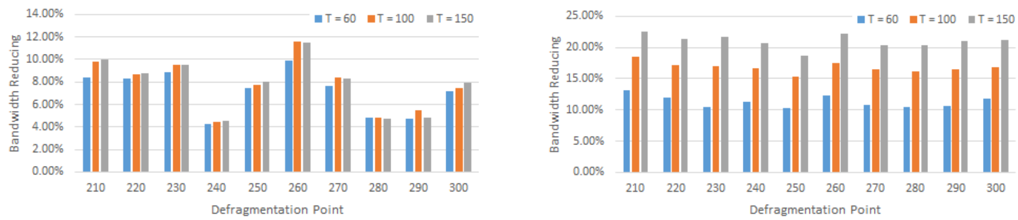
- $\pi_k = 1$ if $k \in K$ is selected for rerouting, 0 otherwise.
- $\alpha_{k\ell} = 1$ if $\pi_k = 1$ and the route of $k \in K$ uses link $\ell \in L$, 0 otherwise.

The goal of PP_t is to select a unique request for potential rerouting, the one with a new route of minimum cost (Objective (8)). Constraints (9) take care of identifying the best possible route, using flow constraints, for the unique request that is rerouted, i.e., such that $\pi_k = 1$. Constraints (10) make sure that we only output simple path, with no loops. Constraints (11) make sure that routing variables $\alpha_{k\ell}$ are null if request k is not selected for rerouting in PP_t. Constraints (12) ensure that PP_t selects only one connection to be rerouted. Observe that we can decompose each PP_t into $|K|$ elementary pricing problems PP_t^k, each examining the option of rerouting request $k \in K$ at time period t by setting $\pi_k = 1$ in PP_t. Let $\bar{c}_{\mathrm{PP}_t}^k$ be the reduced cost of PP_t^k. The solution of PP_t is given by $\bar{c}_{\mathrm{PP}_t} = \min_{k \in K} \{\bar{c}_{\mathrm{PP}_t}^k : k \text{ is rerouted at time period } t\}$.

Now, observe that PP_t^k is a weighted shortest *simple* path problem in a graph with possibly negative weight cycles. This problem is NP-hard (reduction from longest simple path), and so cannot be solved using only the Bellman-Ford-Moore (BFM) algorithm. Indeed, we can still use the BFM algorithm, but with some additional tool. As we cannot enforce the simple path condition, the BFM algorithm may fail to output a path with a negative reduced cost, due to the discovery of a negative cycle. In such a case, we then recourse to the solution of the ILP formulation of PP_t^k ((8)-(14) with $\pi_k = 1$), which includes constraints to enforce the simple path condition. While we did encounter negative loops, it was quite rare and the solution of pricing problems with an ILP solver did not hamper much the overall computational times. It is worth noting that calls to the BFM algorithm are combined by sources. So $|V|$ calls to BFM suffice to solve PP_t.

3 Numerical results

We consider a network with 32 nodes and 250 directed links, which corresponds approximately to a Ciena customer network. Existing network connections were used to construct a traffic matrix input to a simulation generating realistic random connection states. Connection requests had Poisson arrivals based on the traffic matrix and random durations drawn from a common exponential distribution. Each connection had a Weibull distributed bandwidth with a coefficient of variation of 0.3. Connections were routed on the shortest path with sufficient bandwidth. A load factor parameter was used to globally vary the connection arrival rates. For each load factor, we considered 10 defragmentation events. Defragmentation was performed with a period of 1 mean connection duration. Characteristics of the data sets are described in Table 1, where for each load factor, we provide the average number of granted requests right before each defragmentation.



(a) Load factor = 0.5 (b) Load factor = 1.0

FIGURE 2: Reduction (%) of bandwidth requirement

Comparison with the Model of Klopfenstein [4]. We compared the performance of our model and algorithm with the model of Klopfenstein on a dataset with a load factor of 0.5. We report in Table 2 the results for each of the 10 defragmentation events. We observe that although the bandwidth savings obtained with the model of Klopfenstein are better than using DEFRAG_RC (1% in average), the computing times are significantly larger, indeed about 100 times longer. Moreover, the computing time ratio is increasing with the load factor, but due to a lack of space, we only report the results for a 0.5 load factor.

Load factor	Number of requests
0.5	777.2
0.6	894.6
0.7	951.4
0.8	971.1
0.9	993.3
1.0	1,015.4

TABLE 1: Characteristics of the data sets

Defrag.	Bandwidth saving		Computing times	
event	DEFRAG_RC	[4]	DEFRAG_RC	[4]
1	3.2%	3.5%	0.1	13.5
2	7.5%	8.2%	0.2	11.8
3	9.9%	11.0%	0.1	10.2
4	2.3%	2.7%	0.2	29.0
5	9.6%	10.9%	0.1	22.7
6	12.0%	13.7%	0.3	17.1
7	6.6%	7.1%	0.2	15.9
8	3.0%	3.1%	0.1	17.4
9	6.7%	7.3%	0.1	11.9
10	5.4%	5.7%	0.1	2.9
Average	6.6%	7.3%	0.15	15.2
Ratio	1.1		101.3	

TABLE 2: Comparative results with the model of [4]

Defragmentation Performance. We investigated the reduction of the overall bandwidth requirements after each defragmentation, and report in Figure 2 the reduction at each defragmentation event for the two extreme loading factors, i.e., 0.5 and 1.0. Increasing $|T|$ for the 0.5 load does not help to reduce further the bandwidth requirement. For the 1.0 load, we get around an additional 5% bandwidth requirement reduction when increasing $|T|$ from 60 to 100, and then again from 100 to 150.

Références

[1] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse. Tradeoffs in routing reconfiguration problems. In *AlgoTel*, pages 1–4, Belle Dune, France, 2010.

[2] D. Coudert, S. Pérennes, Q.-C. Pham, and J.-S. Sereni. Rerouting requests in WDM networks. In *AlgoTel*, pages 17–20, Presqu'île de Giens, France, May 2005.

[3] B. G. Józsa and M. Makai. On the solution of reroute sequence planning problem in MPLS networks. *Computer Networks*, 42(2) :199–210, 2003.

[4] O. Klopfenstein. Rerouting tunnels for MPLS network resource optimization. *European Journal of Operational Research*, 188(1) :293–312, 2008.

[5] H. Li and J. Wu. Survey of WDM network reconfiguration : topology migrations and their impact on service disruptions. *Telecommunication Systems*, 60 :349–366, Nov. 2015.

[6] F. Solano and M. Pióro. Lightpath reconfiguration in WDM networks. *IEEE/OSA Journal of Optical Communications and Networking*, 2(12) :1010 – 1021, Dec. 2010.