



HAL
open science

Minimum Disturbance Rerouting to Optimize Bandwidth Usage

Huy Duong, Brigitte Jaumard, David Coudert

► **To cite this version:**

Huy Duong, Brigitte Jaumard, David Coudert. Minimum Disturbance Rerouting to Optimize Bandwidth Usage. ONDM 2021 - International Conference on Optical Network Design and Modeling, Jun 2021, Gothenburg, Sweden. pp.1-6, 10.23919/ONDM51796.2021.9492388 . hal-03311598

HAL Id: hal-03311598

<https://inria.hal.science/hal-03311598v1>

Submitted on 1 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Minimum Disturbance Rerouting to Optimize Bandwidth Usage

Huy Duong¹, Brigitte Jaumard², and David Coudert³

¹Centre de Recherche Informatique de Montréal (CRIM), Montréal, Qc, Canada

²Department of Computer Science and Software Engineering, Concordia University,
Montreal (QC) Canada

³Université Côte d’Azur, Inria, CNRS, I3S, France

Abstract

Dynamic traffic leads to bandwidth fragmentation, which drastically reduces network performance, resulting in increased blocking rate and reduced bandwidth usage. When rerouting traffic flows at Layer 3 of an optical network, network operators are interested in minimizing the disturbances in order to satisfy their Service Level Agreements. Therefore, they turn to the Make-Before-Break (MBB) paradigm.

In this paper, we revisit MBB rerouting with the objective of identifying the reroute sequence planning that minimizes the number of reroutes in order to minimize the resource usage. We propose a Dantzig-Wolfe decomposition mathematical model to solve this complex rerouting problem. We instigate how multiple or parallel rerouting reduces the overall minimum number of rerouting events (shortest makespan), and achieve the best resource usage. Numerical results bring interesting insights on that question and show a computational time reduction by about one order of magnitude over the state of the art.

Keywords: Network reconfiguration, rerouting, reoptimization, make-before-break.

1 Introduction

Network resource fragmentation is incurred by dynamic traffic and reduces network efficiency [1]. As part of the resource reallocation by rerouting, many studies have looked at optical path (lightpath) rerouting, both in the context of routing and wavelength allocation (RWA) [2] and routing and spectrum allocation (RSA) [3], or IP / VPN rerouting to overcome failures. However, very few authors have studied rerouting at the IP layer (or Layer 3) for the sole purpose of overcoming fragmentation, optimizing the network resources and increasing the throughput. The latter problem was studied in the 90s in the context of MPLS rerouting, see, e.g., [4] or [5], but then no recent study has been devoted to it, despite a few unanswered questions, and a renewed interest in the context of 5G network slicing. Indeed, network slicing provides a key functionality in 5G networks, and offers flexibility in creating customized virtual networks and supporting different services on a common physical infrastructure [6].

To reroute a connection, the network operator first finds a new route for this connection. To prevent the traffic disruption that occurs when the connection is switched to a new route after the termination of the current route, network operators favor the use of the make-before-break (MBB) paradigm. With MBB, the connection is switched to the new route before the termination of the current route, thus avoiding disruptions.

Recent advances for MBB rerouting [7, 8] enable to significantly improve upon the state of the art on the size of solved instances. However, these proposals did not consider the parallel

and multiple MBB rerouting, i.e., the ability to reroute multiple connections in parallel or to reroute a connection more than once, which could help to further reduce the use of network resources. This approach has only been considered in few studies and the state of the art is the model proposed in [4].

In this paper, we propose an optimization model that takes into account both parallel and multiple MBB rerouting for a capacitated network. We then present decomposition methods based on column generation algorithms to solve this model. The numerical results show that our algorithm is remarkably faster and provides better solutions than [8].

The paper is organized as follows. Section 2 contains the literature review. After a concise problem statement, we propose a decomposition model in Section 3, and its solution process in Section 4. Numerical results are presented in Section 5. Conclusions are drawn in the last section

2 Literature Review

Several studies have been devoted to network reconfiguration with the minimum number of disruptions, following the strategy of migrating from a legacy ineffective provisioning to a given pre-computed optimized/optimal one. As a result, it usually prevents the existence of a strategy using only MBB due to the presence of dependency cycles, also called deadlocks [9]. In order to find a rerouting strategy, authors have then proposed to use the Break-Before-Make (BBM) paradigm sparingly to allow temporary interruption of connection requests, and so to break dependency cycles. For instance, Jose and Somani [10] propose heuristics for minimizing the total number of BBMs used in the rerouting strategy, and Cohen *et al.* [11] and Solano [12] provide scalable exact algorithms to minimize the concurrent number of BBMs and investigate tradeoffs between these two conflicting objectives.

To further reduce the total or concurrent number of BBMs, Kadohata *et al.* [13] propose to use spare wavelengths to reroute a connection request to a temporary route rather than using a BBM. For example, assume that the current connection k needs to be rerouted from path p to path p' , but such a rerouting cannot be MBB due to resource dependence. Then one unavoidable BBM rerouting is performed. However, using an intermediate reroute, it may be possible to reroute k under the MBB paradigm. For instance, assume that there exists a path p'' such that the reroutings from p to p'' and from p'' to p' satisfy the MBB condition. In other words, one BBM can be avoided at the expense of performing two MBBs. This idea is similar to multiple rerouting for capacity reoptimization.

The idea of 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 so-called MBB paradigm. While many studies have investigated the first direction, this second direction has received very little attention [4, 7].

3 Problem Statement

3.1 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 is the set of fiber 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. We denote $\omega^-(v)$ (resp. $\omega^+(v)$) the set of incoming (resp. outgoing) links of node $v \in V$. Let C_ℓ denote the transport capacity of link ℓ .

Let K be the set of connection requests (indexed by k). Connection request $k \in K$ is characterized by its source s_k , its destination d_k , and its bandwidth requirement b_k . In what follows, we call *rerouting operation* the action of rerouting a connection request $k \in K$, and *rerouting event* the action of either performing a single rerouting operation, or a set of parallel rerouting operations. A *reoptimization event* is an ordered sequence of rerouting events, and so of rerouting operations. Let T be the set of rerouting events, indexed by t .

A life-line configuration (or LL config. for short) for request k , denoted by γ , is defined by an ordered sequence of rerouting operations of a connection k , which is characterized by:

- λ_t^γ is 1 if k is rerouted at rerouting event t , 0 otherwise.
- $\alpha_{\ell t}^\gamma$ is 1 if k uses link ℓ at the end of rerouting event t .
- $\delta_{\ell t}^\gamma = \alpha_{\ell t-1}^\gamma - \alpha_{\ell t}^\gamma$ is the difference in usage of link ℓ between the begin of rerouting event t and its end.
- $x_{\ell t}^\gamma$ is 1 if k does not use link ℓ after rerouting event $t - 1$ and uses link ℓ after rerouting event t .

Denote by Γ_k the set of all feasible life-line configurations for connection $k \in K$ with at most R^{MUL} rerouting operations.

3.2 Multiple Parallel MBB Reoptimization Model

Assuming that all feasible life-line configurations are enumerated, the multiple parallel MBB reoptimization model (MUL_PAR_RO) is defined formally by the following decomposition mathematical model.

We first define its parameters:

- R = limit on the total number of rerouting operations.
- R^\parallel = limit on the number of parallel rerouting operations at a rerouting event.
- R^{MUL} = limit on the number of rerouting operations for connection k .
- $a_{k\ell}^{\text{INIT}}$ = 1 if link $\ell \in L$ is used in the initial routing of connection request $k \in K$, 0 otherwise.
- $C_\ell^{\text{INIT}} = \sum_{k \in K} b_k a_{k\ell}^{\text{INIT}}$ = initial load of link $\ell \in L$.

It also uses the following variables:

- $y_\gamma = 1$ if life-line $\gamma \in \Gamma$ is selected, 0 otherwise.
- $C_\ell^t =$ bandwidth usage on link $\ell \in L$ after rerouting event $t \in T$.

Minimize

$$\sum_{\ell \in L} C_\ell^{|T|} \tag{1}$$

subject to:

$$\sum_{k \in K} \sum_{\gamma \in \Gamma_k} \lambda_t^\gamma y_\gamma \leq R^\parallel \quad t \in T \quad (2)$$

$$\sum_{\gamma \in \Gamma} \left(\sum_{t \in T} \lambda_t^\gamma \right) y_\gamma \leq R \quad (3)$$

$$\sum_{\gamma \in \Gamma_k} y_\gamma \leq 1 \quad k \in K \quad (4)$$

$$C_\ell^t \leq C_\ell \quad \ell \in L, t \in T \quad (5)$$

$$C_\ell^{\text{INIT}} = \sum_{k \in K} b_k a_{k\ell}^{\text{INIT}} \quad \ell \in L \quad (6)$$

$$C_\ell^{t-1} + \sum_{k \in K} \sum_{\gamma \in \Gamma_k} b_k \lambda_t^\gamma x_{\ell t}^\gamma y_\gamma \leq C_\ell \quad \ell \in L, t \in T \quad (7)$$

$$C_\ell^{t-1} + \sum_{k \in K} \sum_{\gamma \in \Gamma_k} b_k \lambda_t^\gamma \delta_{\ell t}^\gamma y_\gamma \leq C_\ell^t \quad \ell \in L, t \in T \quad (8)$$

$$C_\ell^t \geq 0 \quad \ell \in L, t \in T \quad (9)$$

$$y_\gamma \in \{0, 1\} \quad k \in K, \gamma \in \Gamma_k \quad (10)$$

The objective (1) is to minimize the capacity usage at the end of the reoptimization event. Constraints (2) prevent the selection of more than R^\parallel rerouting operations at each rerouting event. Constraints (3) restrict the total number of rerouting operations of the whole reoptimization event. In case one does not want to restrict the number of parallel rerouting operations per rerouting event, R^\parallel is set to infinite and Constraints (2) are removed from the model. Note that, in case $R^{\text{MUL}} = \infty$, $|T| \leq |K|$ is an upper bound on the number of rerouting operations.

Constraints (4) ensure that a connection request has at most one life-line. Constraints (5) make sure that transport capacities are never exceeded after any rerouting event.

Constraints (6) specifies the initial bandwidth usage of each link. Constraints (7) ensure that the bandwidth which is needed on link ℓ for the "make" part does not exceed its capacity at rerouting event t . Note that if the old and new routes of a connection go through a same link, reserved capacity on that link is not duplicated. Constraints (8) update the bandwidth usage on link ℓ after rerouting event t . Constraints (9)-(10) define the domain of the variables.

4 Solution Process

4.1 Life-Line Pricing Algorithm - CPP

We use a column generation algorithm to solve the linear relaxation of Model (1)-(10) (see, e.g., [14] if not familiar with column generation techniques). It results in decomposing the original problem into a Restricted Master Problem (RMP), i.e., Model (1)-(10) with a very restricted number of variables, and one or several pricing problems (PPs), which are solved alternately. The process continues until the optimality condition is satisfied, i.e., all the so-called reduced costs (or equivalently, the optimal values of the objective functions of the pricing problems) are positive. An ε -optimal solution is derived by solving exactly the ILP model associated with the last RMP, with ε defined as follows:

$$\varepsilon = (\tilde{z}_{\text{ILP}} - z_{\text{LP}}^*) / z_{\text{LP}}^*, \quad (11)$$

where z_{LP}^* and \tilde{z}_{ILP} denote the optimal LP value and the optimal ILP value of the last RMP, respectively.

We now describe the pricing problem model. As the parameters of a life-line configuration become variables in the pricing problem, we do not change their names (slight abuse of notations).

Let $u_t^{(2)} \leq 0$, $u_{\ell t}^{(3)} \leq 0$, $u_k^{(4)} \leq 0$, $u_{\ell t}^{(7)} \leq 0$, and $u_{\ell t}^{(8)} \leq 0$ be the dual values of Constraints (2),(3), (4), (7), and (8) respectively.

Variables:

- $\alpha_{\ell t}$ is 1 if the routing of connection k uses link $\ell \in L$ at the end of rerouting event t , 0 otherwise.
- λ_t is 1 if t is a rerouting event, i.e. if a rerouting operation occurs, 0 otherwise.
- $x_{\ell t}$ is 1 if link $\ell \in L$ is used at the end of rerouting event t but was not used at $t - 1$, 0 otherwise.
- $\delta_{\ell t}$ is 1 if link $\ell \in L$ is used at t but not at $t - 1$, -1 if it was used at $t - 1$ and is no longer used, and 0 if its usage is unchanged.
- $p_v^t \geq 0$ encodes the relative position of vertex $v \in V$ in the path at t . It is used for the elimination of subtours.

Objective (reduced cost):

$$[\text{PP}_k] \equiv \min \quad \bar{c}_{\text{PP}_k} = - \sum_{t \in T} (u_t^{(2)} + u^{(3)}) \lambda_t - u_k^{(4)} - \sum_{\ell \in L} \sum_{t \in T} u_{\ell t}^{(7)} b_k \lambda_t x_{\ell t} - \sum_{\ell \in L} \sum_{t \in T} b_k u_{\ell t}^{(8)} \lambda_t \delta_{\ell t} \quad (12)$$

Subject to:

$$\sum_{\ell \in \omega^-(v)} \alpha_{\ell t} - \sum_{\ell \in \omega^+(v)} \alpha_{\ell t} = \begin{cases} -1 & \text{if } v = s_k \\ 1 & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases} \quad t \in T, v \in V \quad (13)$$

$$\sum_{\ell \in \omega^+(v)} \alpha_{\ell t} \leq 1 \quad t \in T, v \in V \quad (14)$$

$$\alpha_{\ell 0} = a_{k\ell}^{\text{INIT}} \quad \ell \in L \quad (15)$$

$$\sum_{t \in T} \lambda_t \leq R^{\text{MUL}} \quad (16)$$

$$\lambda_t \geq \alpha_{\ell, t-1} - \alpha_{\ell, t} \quad t \in T, \ell \in L \quad (17)$$

$$x_{\ell t} \geq \alpha_{\ell, t} - \alpha_{\ell, t-1} \quad t \in T, \ell \in L \quad (18)$$

$$x_{\ell t} \leq \lambda_t \quad t \in T, \ell \in L \quad (19)$$

$$\sum_{\ell \in L} x_{\ell t} \geq \lambda_t \quad t \in T \quad (20)$$

$$\delta_{\ell t} = \alpha_{\ell, t} - \alpha_{\ell, t-1} \quad t \in T, \ell \in L \quad (21)$$

$$p_v^t \geq p_u^t + 1 + |V|(\alpha_{\ell, t} - 1) \quad t \in T, (u, v) \in L \quad (22)$$

Domains of the variables:

$$\alpha_{\ell t} \in \{0, 1\} \quad t \in T \cup \{0\}, \ell \in L \quad (23)$$

$$\lambda_t \in \{0, 1\} \quad t \in T \quad (24)$$

$$x_{\ell t} \in \{0, 1\}, \quad t \in T, \ell \in L \quad (25)$$

$$\delta_{\ell t} \in \{-1, 0, 1\} \quad t \in T, \ell \in L \quad (26)$$

$$p_v^t \geq 0 \quad t \in T, v \in V. \quad (27)$$

The objective function (12) is quadratic. However, the first quadratic term $\lambda_t x_{\ell t}$ can be equivalently rewritten $x_{\ell t}$ thanks to Constraints (19), and the second quadratic term $\lambda_t \delta_{\ell t}$ can be equivalently rewritten $\delta_{\ell t}$ thanks to the combination of Constraints (17) and Constraints (21).

The overall set of Constraints (13)-(26) describes the life-line of request k .

Constraints (13) are flow conservation constraints defining a path for each connection after each rerouting event, while avoiding loops along the path thanks to Constraints (14). Constraints (15) specify the links that are used in the initial routing of request k . Constraint (16) limits the number of rerouting operations on the life-line of connection k . Constraints (17)-(20) identify rerouting operations and consequently rerouting events. Let us first assume that $\lambda_t = 0$. Then, Constraints (17) ensure that the links used at $t-1$ are still used at t , and Constraints (18)-(19) prevent from using new links at t . Hence, the routing at $t-1$ and t are the same. Assume now that a rerouting operation occurs at t , that is $\lambda_t = 1$. Constraints (20) ensure that at least one variable $x_{\ell t}$ is set to 1, thus enabling (i) to use at t a link ℓ that was not used at $t-1$ with Constraints (19) (ii) to stop using at t some links that were used at $t-1$ with Constraints (17). On the other hand, if either a link is no longer used at t , or a link is used at t but was not used at $t-1$, Constraints (18)-(19) identify that a rerouting event occurs at t and set variable λ_t to 1. If no new link is used at t , Constraints (20) force variable λ_t to 0, Constraints (17) force to continue using at t the links that were used at $t-1$, and so the paths at $t-1$ and t are the same. Constraints (21) encode in $\delta_{\ell t}$ the changes in link usage. Constraints (22) prevent the selection of cycles. Finally, Constraints (23)-(27) define the domains of the variables.

4.2 Solution Process of Non-Multiple (Single) Rerouting per Connection

When $R^{\text{MUL}} = 1$, each connection $k \in K$ can be rerouted at most once, and so, at most one λ_t can be one. As in Duong *et al.* [8], we can decompose the compact pricing problem into a set of pricing problems $\text{PP}_{kt}^{\text{SR}}$ in which $\lambda_t = 1$. We then get:

$$\bar{c}_{\text{PP}_k^{\text{SR}}} = \min_{t \in T} \bar{c}_{\text{PP}_{kt}^{\text{SR}}} = -u_k^{(4)} - \max_{t \in T} \left(u_t^{(2)} + u^{(3)} + \sum_{\ell \in L} b_k u_{\ell t}^{(7)} x_{\ell t} + \sum_{\ell \in L} b_k u_{\ell t}^{(8)} \delta_{\ell t} \right) \quad (28)$$

Assuming $\lambda_t = 1$, Constraints (17) and (19) become redundant for the selected t , and can therefore be eliminated. The simplified pricing problem $\text{PP}_{kt}^{\text{SR}}$ with single rerouting per connection can be written as follows.

$$[\text{PP}_{kt}^{\text{SR}}] \equiv \bar{c}_{\text{PP}_{kt}^{\text{SR}}}^* \equiv \min \quad \bar{c}_{\text{PP}_{kt}^{\text{SR}}} = -u_t^{(2)} - u^{(3)} - u_k^{(4)} \\ + \sum_{\ell \in L} b_k u_{\ell t}^{(8)} a_{k\ell}^{\text{INIT}} - b_k \sum_{\ell \in L} (u_{\ell t}^{(7)} x_{\ell t} + u_{\ell t}^{(8)} \alpha_{\ell t}) \quad (29)$$

subject to:

$$\sum_{\ell \in \omega^-(v)} \alpha_{\ell t} - \sum_{\ell \in \omega^+(v)} \alpha_{\ell t} = \begin{cases} -1 & \text{if } v = s_k \\ 1 & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases} \quad v \in V \quad (30)$$

$$\sum_{\ell \in \omega^+(v)} \alpha_{\ell t} \leq 1 \quad v \in V \quad (31)$$

$$x_{\ell t} \geq \alpha_{\ell t} - 0 \quad \ell \in L_k : \alpha_{\ell, t-1} = 0 \quad (32)$$

$$x_{\ell t} \geq \alpha_{\ell t} - 1 \quad \ell \in L_k : \alpha_{\ell, t-1} = 1 \quad (33)$$

$$\sum_{\ell \in L} x_{\ell t} \geq 1 \quad (34)$$

$$p_v \geq p_u + 1 + |V|(\alpha_{\ell, t} - 1) \quad (u, v) = \ell \in L \quad (35)$$

$$\alpha_{\ell t} \in \{0, 1\}, \quad x_{\ell t} \in \{0, 1\} \quad \ell \in L \quad (36)$$

$$p_v \geq 0 \quad v \in V. \quad (37)$$

Note that in this pricing problem, when $t - 1 = 0$, we have $\alpha_{\ell, t-1} = a_{k\ell}^{\text{INIT}}$.

Proposition 1. *If we are looking for only negative optimal objective value (reduced cost), pricing problem $\text{PP}_{kt}^{\text{SR}}$ can be reduced to a shortest path problem with non negative weights.*

Proof. Firstly, we show that variables x can be eliminated. Let

$$u_{\ell t}^{(7)(8)} = \begin{cases} u_{\ell t}^{(8)} & \text{if } a_{k,\ell}^{\text{INIT}} = 1 \\ u_{\ell t}^{(7)} + u_{\ell t}^{(8)} & \text{if } a_{k,\ell}^{\text{INIT}} = 0 \end{cases} \quad (38)$$

and consider the following model.

$$\bar{c}_{\text{PP}_{kt}^{\text{SR}}}^* \equiv \min \quad \bar{c}_{\text{PP}_{kt}^{\text{SR}}} = -u_t^{(2)} - u^{(3)} - u_k^{(4)} + \sum_{\ell \in L} b_k u_{\ell t}^{(8)} a_{k,\ell}^{\text{INIT}} - \sum_{\ell \in L} u_{\ell t}^{(7)(8)} b_k \alpha_{\ell, t} \quad (39)$$

subject to Constraints (30)-(37)

We will prove that the model using Objective (39) is equivalent to the original pricing model of $\text{PP}_{kt}^{\text{SR}}$ using Objective (29). To do that, we first prove the two following claims.

- (i) The feasible regions of these models are equal. This claim is trivial because these problems have the same sets of constraints.
- (ii) The optimal solutions of these models are equal. This claim will be proved if the two following statements are true:

$$\text{(ii-a)} \quad \bar{c}_{\text{PP}_{kt}^{\text{SR}}}^* \geq \bar{c}_{\text{PP}_{kt}^{\text{SR}}}^*;$$

$$\text{(ii-b)} \quad \exists (x', \alpha') \in \{(30) - (37)\} : \bar{c}_{\text{PP}_{kt}^{\text{SR}}}^* = \bar{c}_{\text{PP}_{kt}^{\text{SR}}}(x', \alpha').$$

To prove statement (ii-a), we must show that it is true for any valid assignment of the variables, that is:

$$\forall (x', \alpha') \quad \bar{c}_{\text{PP}_{kt}^{\text{SR}}}(x', \alpha') \geq \bar{c}_{\text{PP}_{kt}^{\text{SR}}}^*(x', \alpha'). \quad (40)$$

Since $u_{\ell t}^{(7)} \leq 0$ and $u_{\ell t}^{(8)} \leq 0$, this statement holds if

$$\forall \ell \in L \quad -u_{\ell t}^{(7)} x'_{\ell t} - u_{\ell t}^{(8)} \alpha'_{\ell t} \geq -u_{\ell t}^{(7)(8)} \alpha'_{\ell t}$$

is true, and so equivalently, if the following statement is true.

$$\forall \ell \in L \quad u_{\ell t}^{(7)} x'_{\ell t} + u_{\ell t}^{(8)} \alpha'_{\ell t} \leq u_{\ell t}^{(7)(8)} \alpha'_{\ell t}.$$

Indeed,

- If $\ell \in L$ is such that $a_{k,\ell}^0 = 0$, we have by Constraints (32) that $x'_{\ell t} \geq \alpha'_{\ell t}$. Since $u_{\ell t}^{(7)} \leq 0$, we get $u_{\ell t}^{(7)} x'_{\ell t} + u_{\ell t}^{(8)} \alpha'_{\ell t} \leq u_{\ell t}^{(7)} \alpha'_{\ell t} + u_{\ell t}^{(8)} \alpha'_{\ell t}$, and Equation (38) sets $u_{\ell t}^{(7)(8)} = u_{\ell t}^{(7)} + u_{\ell t}^{(8)}$.
- If $\ell \in L$ is such that $a_{k,\ell}^0 = 1$, we have by Constraints (33) that $x'_{\ell t} \geq \alpha'_{\ell t} - 1$. Since $x'_{\ell t} \in \{0, 1\}$ and $u_{\ell t}^{(7)} \leq 0$, we therefore have $u_{\ell t}^{(7)} x'_{\ell t} \leq 0$. Since Equation (38) sets $u_{\ell t}^{(7)(8)} = u_{\ell t}^{(7)} + u_{\ell t}^{(8)}$, we can conclude that $u_{\ell t}^{(7)} x'_{\ell t} + u_{\ell t}^{(8)} \alpha'_{\ell t} \leq u_{\ell t}^{(7)(8)} \alpha'_{\ell t}$.

We now prove statement (ii-b). Let x^* and α^* be the optimal solution corresponding to $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^*$. We show that if $\alpha' = \alpha^*$, and x' such that

$$x'_{\ell t} = \begin{cases} 0 & \text{if } a_{k,\ell}^0 = 1 \\ \alpha^*_{\ell t} & \text{if } a_{k,\ell}^0 = 0 \end{cases}, \quad (41)$$

then $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* = \overline{c}_{\text{PP}_{kt}^{\text{SR}}}(\alpha^*, x^*) = \overline{c}_{\text{PP}_{kt}^{\text{SR}}}(\alpha', x')$.

First, we need to show that α' and x' are a feasible solution of (30)-(37). Clearly, α' satisfies (30)-(31) because $\alpha' = \alpha^*$. Furthermore, Constraints (32)-(33) are satisfied by α' and x' because they hold in all cases of $\ell \in L$. Indeed, we have

- If $\ell \in L$ is such that $a_{k,\ell}^{\text{INIT}} = 0$, then we have $x'_{\ell t} = \alpha^*_{\ell t} = \alpha'_{\ell t} \geq \alpha'_{\ell t} - 0$.
- If $\ell \in L$ is such that $a_{k,\ell}^{\text{INIT}} = 1$, then we have $x'_{\ell t} = 0 \geq \alpha'_{\ell t} - 1$ (as $\alpha'_{\ell t} \in \{0, 1\}$).

Note that we only consider the original pricing when $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* \leq 0$. In that case, there has to exist one $\ell \in L$ such that $a_{k,\ell}^{\text{INIT}} = 0$ and $\alpha^*_{\ell t} > 0$. Otherwise, $\alpha^*_{\ell t} = a_{k,\ell}^{\text{INIT}}$ for all $\ell \in L$ (the only feasible path in this case), then

$$\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* = -u_t^{(2)} - u^{(3)} - u_k^{(4)} > 0 \implies \overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* \geq \overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* > 0,$$

this contradicts the assumption that $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* \leq 0$. Overall, it concludes that, in this case, Constraint (34) is satisfied by x' .

When $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* > 0$, it implies $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}^* > 0$, so the pricing problem cannot generate improving configuration for the restricted master problem.

We now show that the objective values are equal. Note that $\overline{c}_{\text{PP}_{kt}^{\text{SR}}}$ does not depend on x^* , and so is computed with α^* only. Now, we show that

$$\forall \ell \in L \quad u_{\ell t}^{(7)(8)} \alpha^*_{\ell t} = u_{\ell t}^{(7)} x'_{\ell t} + u_{\ell t}^{(8)} \alpha'_{\ell t}. \quad (42)$$

This holds trivially by the combination of (38) and (41).

To this end, we observe that variables x can be removed from the simplified problem and it becomes a shortest path problem with non negative weights. \square

5 Numerical Results

5.1 Data Sets

We use the instances resulting from the simulation performed in [8] on a network with 32 nodes and 250 directed links, which corresponds approximately to a Ciena customer network. Connection requests had Poisson arrivals based on a 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. A load factor parameter was used to globally vary the connection arrival rates: the corresponding equilibrium connection states represent a range of congestion levels from light to heavy.

All experiments are run with Scientific Linux release 7.9 (Nitrogen), Intel(R) Xeon(R) CPU E7-4890 v2 @ 2.80GHz, 15 cores per socket, 2 threads per core, and 1056 GB of RAM.

5.2 Regular Multiple Parallel Rerouting CG Performance

We now evaluate the performances of the CPP algorithm in data instances with 12 nodes.

In Table 1, we report experiments when at most one rerouting operation is allowed per request ($R^{\text{MUL}} = 1$) while the number of rerouting operations per rerouting event is unlimited. The average gap of the CPP algorithm over 10 reoptimization events is 1.3%. It shows that the proposed algorithm is close to the optimal solution in this setting. We also observe that the time required to solve the last ILP problem is negligible in the overall resolution time of the CPP algorithm, meaning that the difficult task of the proposed algorithm is the generation of life-lines.

Table 1: CPP Algo. Performance, $T = 5$, $R = 15$, $R^{\parallel} = \infty$, $R^{\text{MUL}} = 1$, $|V| = 12$, $|L| = 52$

Scenarios	$ K $	ε (%)	# LP LLs	ILP CPU (s)	\tilde{z}_{ILP}	# rerouting op.	# ILP LLs	Init BW	z_{LP}^*	CPU (s)
1	168	1.9	637	0.1	165,765	11	11	176,879	162,721.0	65.3
2	184	0.5	338	0.0	180,652	15	15	197,224	179,751.0	32.6
3	190	1.3	381	0.0	179,538	15	15	196,782	177,264.1	37.2
4	191	3.7	653	0.1	189,029	12	12	200,844	182,311.4	55.4
5	180	0.1	321	0.0	175,989	15	15	196,493	175,751.4	48.7
6	181	0.9	386	0.0	172,824	9	9	182,751	171,357.0	39.1
7	177	0.6	509	0.1	171,347	15	15	188,211	170,345.0	44.2
8	184	0.6	449	0.0	178,750	12	12	191,574	177,726.0	52.4
9	180	2.5	649	0.1	184,146	13	13	202,182	179,693.9	62.7
10	167	0.5	382	0.0	168,583	15	15	190,894	167,677.6	43.5
Average	180.2	1.3	470.5	0.1	176662.3	13.2	13.2	192383.4	174459.8	48.1

In Table 2, we report experiments when two rerouting operations are allowed per request (i.e., $R^{\text{MUL}} = 2$) and the number of rerouting operations per rerouting event is unlimited. Before solving the last ILP problem of this case, we use the solutions of the experiments with $R^{\text{MUL}} = 1$ as additional columns to generated columns. The rows in Table 2 where the number of routing operations is larger than the number of selected life-line configurations (# ILP LLs) indicate that there are connections that are rerouted twice in the solution.

In Table 2, we compute the bandwidth usage gain when we can reroute a connection at most twice. Gain is not significant (less than 1%) when we allow 2 reroutings per connection for small networks (about 10 nodes), as connections are often initially routed on their shortest paths or close to shortest paths. So, there is very limited room for improvement. Note that, for the fifth rerouting event, the gain is negative. It is because the gap of the fifth rerouting event of Table 2 (0.2%) is worse than the corresponding event of Table 1 (0.1%).

When the number of allowed rerouting operations per request is increased, it leads to more difficult problem and takes more time to converge, as demonstrated by the significant difference

of the number of generated life-lines ("# LP LLs" column) and total CPU time between Tables 1 and 2.

Table 2: CPP Algo. Performance, $T = 5$, $R = 15$, $R^{\parallel} = \infty$, $R^{\text{MUL}} = 2$, $|V| = 12$, $|L| = 52$

Scenarios	$ K $	ε (%)	# LP LLs	ILP CPU (s)	\tilde{z}_{ILP}	Gain	# rerouting op.	# ILP LLs	Init BW	z_{LP}^*	CPU (s)
1	168	1.4	2,114	0.3	164,923	842	13	12	176,879	162,721.0	188.7
2	184	0.1	1,854	0.3	179,939	713	15	15	197,224	179,751.0	523.4
3	190	0.9	2,027	0.3	178,927	611	15	15	196,782	177,264.1	500.7
4	191	2.7	2,624	0.4	187,175	1,854	14	14	200,844	182,311.4	377.1
5	180	0.2	984	0.2	176,029	-40	15	15	196,493	175,751.4	124.1
6	181	0.6	1,500	0.3	172,447	377	11	10	182,751	171,357.0	163.9
7	177	0.6	1,882	0.3	171,347	0	15	15	188,211	170,345.0	230.4
8	184	0.6	2,042	0.4	178,750	0	14	13	191,574	177,726.0	237.8
9	180	2.4	3,007	0.7	184,091	55	15	15	202,182	179,693.9	533.3
10	167	0.2	1,755	0.3	167,986	597	15	15	190,894	167,677.6	648.9
Average	180.2	1.0	1978.9	0.4	176161.4	500.9	14.2	13.9	192383.4	174459.8	352.8

5.3 Non-multiple Rerouting Algorithm

In this section, we evaluate the performance of the CPP algorithm when it is specialized for the particular case where each connection is allowed to be rerouted at most once, i.e., $R^{\text{MUL}} = 1$. As aforementioned in Section 4.2, the life-line generation pricing problem can be solved by decomposing it into a set of simple non-negative weighted shortest path problems, which can be solved using Dijkstra's algorithm.

In the first part of Table 3, we report average performance of specialized CPP algorithm over 10 reoptimization events for each traffic intensity (load) with $T = 20$, $R = 400$, $R^{\text{MUL}} = 1$ and $R^{\parallel} = 20$. The reported performance parameters are as in Table 1 and Table 2. Besides, in this table, "BW reduction" column reports reduced portions of bandwidth requirements before and after rerouting events using proposed CPP algorithm (i.e., $\frac{\tilde{z}_{\text{ILP}} - \text{"Init. BW"}}{\text{"Init. BW"}}$). In the second part of Table 3, the similar results in [8] are recalled for the case where the total number of rerouting operations is 150.

We observe that the CPU times of non-multiple rerouting algorithm is about one order of magnitude smaller than the algorithm of [8], as we use a much smaller number of rerouting events ($T = 20$ instead of $T = 150$ in [8]), while preserving a comparable accuracy to that of [8].

Table 3: Non-multiple Rerouting Algorithm's Performance ($T = 20$, $R = 400$, $R^{\text{MUL}} = 1$, $R^{\parallel} = 20$, $|V| = 32$, $|L| = 250$)

Load	CPP algorithm					Duong <i>et al.</i> [8] (150 allowed rerouting operations)			
	# LP LLs	# ILP LLs	ε (%)	CPU (s)	BW reduction (%)	# Gen. config.	In sol. config.	ε (%)	CPU (min.)
0.5	863.6	71.9	2.2	90.6	7.8	3,465.1	63.5	2.2	89.1
0.6	1313.6	154.5	4.7	132.4	17.6	3,669.5	139.2	2.7	316.3
0.7	1624.3	198.6	5.5	178.6	22.7	2,782.4	147.2	2.3	235.0
0.8	1634.1	218.4	5.0	201.3	24.5	2,798.7	145.9	2.6	230.5
0.9	1728.1	233.5	4.9	207.1	26.1	2,804.2	147.9	2.8	263.5
1	1816.3	247.2	3.9	221.2	26.7	2,806.2	147.5	2.3	264.6

We report the reduction of the overall bandwidth requirements after each reoptimization event for the two extreme loading factors, i.e., 0.5 and 1.0, see Figure 1.

6 Conclusions

We have proposed a new model for flow rerouting with the Make-Before-Break paradigm at Layer 3 in optical networks. It generalizes the previous MBB rerouting problem, not yet discussed in

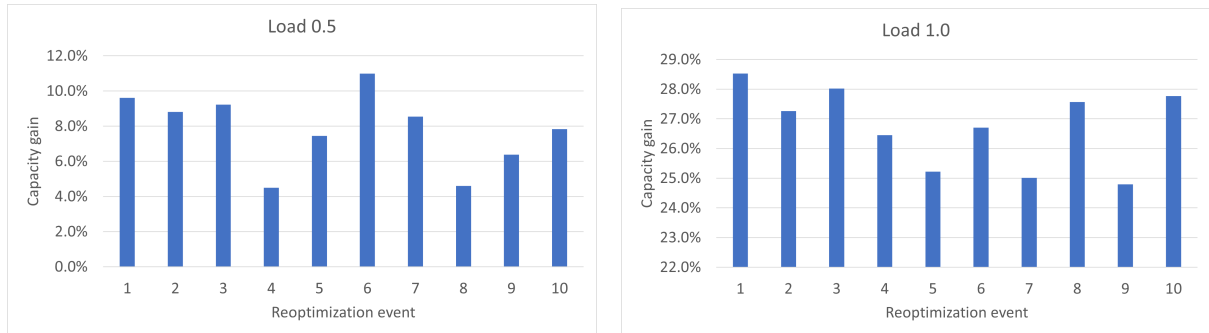


Figure 1: Reduction (%) of capacity requirement for load factors 0.5 and 1.

literature, i.e., with multiple rerouting operations per connection and parallel rerouting operations per rerouting event, while minimizing the overall number of rerouting in order to minimize the number of disturbance.

In addition, this modeling leads to an efficient algorithm to solve the specific case of one rerouting operation per connection: we can perform up to 400 rerouting operations in about 5 minutes, while the model of [8] was only scalable for 150 rerouting operations requiring about one hour.

Acknowledgments

B. Jaumard has been supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant. H. Duong was supported by a MITACS & Ciena Converge Fellowship. D. Coudert has been supported by the French National Research Agency (ANR), through the UCA^{JEDI} Investments in the Future project with the reference number ANR-15-IDEX-0001, and the Inria associated-team project EfDyNet.

References

- [1] B. C. Chatterjee, S. Ba, and E. Oki, “Fragmentation problems and management approaches in elastic optical networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 183 – 210, 2020.
- [2] B. Jaumard, H. Pouya, and D. Coudert, “Wavelength defragmentation for seamless migration,” *Journal of Lightwave Technology*, vol. 37, no. 17, pp. 4382 – 4393, 2019.
- [3] P. Soumplis, K. Christodoulopoulos, and E. Varvarigos, “Dynamic connection establishment and network re-optimization in flexible optical networks,” *Photonic Network Communication*, vol. 29, pp. 307 – 321, June 2015.
- [4] O. Klopfenstein, “Rerouting tunnels for MPLS network resource optimization,” *European Journal of Operational Research*, vol. 188(1), pp. 293 – 312, 2008.
- [5] B. Józsa and M. Makai, “On the solution of reroute sequence planning problem in MPLS networks,” *Computer Networks*, vol. 42, pp. 199 – 210, June 2003.

- [6] S. Mandelli, M. Andrews, S. Borst, and S. Klein, “Satisfying network slicing constraints via 5G MAC scheduling,” in *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM*, 2019, pp. 2332–2340.
- [7] B. Jaumard, H. Quang Duong, R. Armolavicius, T. Morris, and P. Djukic, “Efficient real-time scalable make-before-break network re-routing,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, p. 52, 03 2019.
- [8] H. Duong, B. Jaumard, D. Coudert, and R. Armolavicius, “Efficient Make Before Break Capacity Defragmentation,” in *IEEE International Conference on High Performance Switching and Routing - HPSR*, Bucharest, Romania, June 2018, pp. 1–6.
- [9] L. Maggi, P. Poirion, and J. Leguay, “Reroute backward to better break deadlocks,” in *IEEE International Conference on Cloud Networking (CloudNet)*, Prague, Czech Republic, 2017, pp. 1 – 6.
- [10] N. Jose and A. K. Somani, “Connection rerouting/network reconfiguration,” in *Proceedings of IEEE/VDE Workshop on Design of Reliable Communication Networks - DRCN*, 2003, pp. 23–30.
- [11] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse, “Tradeoffs in process strategy games with application in the WDM reconfiguration problem,” *Theoretical Computer Science*, vol. 412, no. 35, pp. 4675–4687, 2011.
- [12] F. Solano, “Analyzing two conflicting objectives of the WDM lightpath reconfiguration problem,” in *IEEE Global Telecommunications Conference - GLOBECOM*, Nov. 2009, pp. 1–7.
- [13] A. Kadohata, A. Hirano, F. Inuzuka, A. Watanabe, and O. Ishida, “Wavelength path reconfiguration design in transparent optical WDM networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 7, pp. 751 – 761, July 2013.
- [14] V. Chvatal, *Linear Programming*. Freeman, 1983.