



**HAL**  
open science

# $k$ -Adaptive Routing for the Robust Network Loading Problem

Marco Silva, Michael Poss, Nelson Maculan

► **To cite this version:**

Marco Silva, Michael Poss, Nelson Maculan.  $k$  -Adaptive Routing for the Robust Network Loading Problem. *Electronic Notes in Discrete Mathematics*, 2018, 64, pp.95-104. 10.1016/j.endm.2018.01.011 . hal-01768637

**HAL Id: hal-01768637**

**<https://hal.science/hal-01768637v1>**

Submitted on 30 Mar 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.

# $k$ -Adaptive Routing for the Robust Network Loading Problem

Marco Silva<sup>a,c,1</sup>, Michael Poss<sup>b,2</sup>, Nelson Maculan<sup>c,3</sup>

<sup>a</sup> *LIA, Université d'Avignon et des Pays des Vaucluse, Avignon, France*

<sup>b</sup> *UMR CNRS 5506 LIRMM, Université de Montpellier, Montpellier, France*

<sup>c</sup> *PESC, UFRJ, Rio de Janeiro, Brazil*

---

## Abstract

We experiment an alternative routing scheme for the Robust Network Loading problem. Named  $k$ -adaptive, it is based on the fact that the decision-maker chooses  $k$  second-stage solutions and then commits to one of them only after realization of the uncertainty. This routing scheme, with its corresponding  $k$ -partition of the uncertainty set, is dynamically defined under an iterative method to sequentially improve the solution. The method has an inherent characteristic of multiplying the number of variables and constraints after each iteration, so that additional measures are introduced in the solution strategy in order to control its tractability. We compare our  $k$ -adaptive results with the ones obtained through other routing schemes and also verify the effectiveness of the methods utilized using several realistic instances from SNDlib.

*Keywords:* Robust Linear Optimization, Budget Uncertainty Set,  $k$ -Adaptability, Network Loading Problem

---

<sup>1</sup> Email: marcosil@cos.ufrj.br

<sup>2</sup> Email: michael.poss@lirmm.fr

<sup>3</sup> Email: maculan@cos.ufrj.br

# 1 Introduction

This study is about a robust linear optimization approach for the network loading problem under demand uncertainty (RNL). Given a graph  $G(V, E)$  and a set of point-to-point uncertain demands as commodities origin-destination flows, we want to define minimum cost capacity installations for the edges (investment decisions), such that all commodities can be routed simultaneously on the network (routing decisions defining a routing scheme).

RNL has a intrinsic separation between first stage decisions and second stage decisions: investment decisions must be made before we observe the results of the demand uncertainty, while the routing decisions made by the decision maker have to route whatever demand occurred. One can decide which route to take based on actual demands and the design of the demand uncertainty impacts the quality and robustness of the solution.

Uncertainty in demands and the question of how to adapt commodities flows to different realizations of demand lead to the concept of routing schemes. Two extremes routing schemes are defined as static (oblivious) and dynamic. Static routing means that for every commodity the same paths are used with the same splitting independently of the realization of demand. On the contrary, dynamic routing permits full flexibility in re-routing according to demand changes and, in consequence, is potentially less costly.

It has been shown by [9] that the robust network design with dynamic routing is NP-hard for polyhedral uncertainty. This difficulty has been addressed in the literature (see [11] and [14]) by restricting the second stage variables to be affine functions of the uncertain data, in a routing scheme called affine adaptability that provides intermediary solutions between static and dynamic. An alternative routing scheme that can be considered, studied in [2], is the  $k$ -adaptive routing, where the decision-maker chooses  $k$  second-stage solutions, and then commits to one of them only after seeing the realization of the uncertainty. The decision-maker selects a set of a partition of the uncertainty set among  $k$  (possibly non-disjoint) regions (see also [1] and [13]).

The inequalities  $Static(P) \geq kAdapt(P) \geq Dynamic(P)$ , meaning a comparison between different routings solutions for the same problem  $P$ , hold in general. On the other hand, in [2], the authors affirm that their  $k$ -adaptability proposal is not comparable to affine adaptability: in some cases affine adaptability fails where  $k$ -adaptability succeeds, and vice versa. The quality of solutions obtained with a  $k$ -adaptive approach depends on how the partition is built. Therefore there is a natural trade-off between the number of sets of a partition (and the solution time) and how “close” the solution is to the

dynamic solution. Hence, a goal when using  $k$ -adaptability is to identify a partitioning scheme that is near the efficient frontier of this trade-off.

In [3] and [15] the authors independently introduce strategies to re-optimize the solution based on a new  $k$ -partitioning of the uncertainty set. The strategies include the following main steps:

- They analyze the optimal solution to a static version of a problem to gain insight into which regions of the uncertainty set are restricting the objective function value.
- The information above is used to construct partitions of the uncertainty set, leading to a  $k$ -adaptable formulation of the problem with potentially improved objective function value.
- This process is iterative and further improves the objective until a bound gap or other termination criterion is reached. An equivalent static robust problem is resolved at each iteration.

The pseudo code below reflects these main steps and defines the algorithm implemented in our experiment. Each step is further explained in the solution strategy below.

---

**Algorithm 1** Iterative  $k$ -Adaptive routing

---

```

Initialize                                     ▷ No partition at first round
repeat
  Solve Equivalent Static RNL Problem
  Calculate Bound
  if Bound gap  $\geq$  Threshold then
    Calculate Active Uncertain Parameters
    Formulate Nested Partitioning
  end if
until Termination Criteria reached

```

---

## 2 Problem definition

We work with two cases: one in which flows are unsplittable, or nonbifurcated, and must use a single path, and another one in which flows are splittable, or bifurcated, and can be fractionally split along several paths. The cost for routing flows is zero. Each commodity  $q \in Q$  is associated with the uncertain demand  $d^q$ , within a given uncertainty set. Specifically the demand for commodity  $q \in Q$  varies around a given nominal demand  $\bar{d}^q \geq 0$  with a

maximal possible deviation of  $\hat{d}^q \geq 0$ , that is,  $d^q(\xi) = \bar{d}^q + \xi^q \hat{d}^q$  for  $\xi \in \Xi$  where  $\Xi \subseteq [0, 1]^q$ .

We focus on a simplification of RNL where each commodity  $q$  can be routed along a predetermined set of paths  $P(q)$ . In many practical applications there are limits on the number of different configurations of paths that can be implemented, so that we exercise the flexibility of routing schemes to mitigate the static solution conservativeness even with this simplification. It has been used in several papers, see for instance [11] and [12]. These paths are represented by  $\delta_{qp}^{ij}$  that is equal to 1 if edge  $ij$  is contained in path  $p \in P(q)$ , for some  $q \in Q$  and is equal to 0 otherwise. They were predetermined as many shortest paths weighted by edges costs for each commodity.

The formulation contains integer investment decision variables  $x$ , where  $x_{ij}$  equals the planned installed capacity for edge  $ij$ , at a unit cost of  $c_{ij}$ . considering only one type of facility. It contains continuous (for bifurcated flows) or binary (for nonbifurcated flows) routing decision variables  $y_{qp}$ , where  $y_{qp}$  equals the fraction of commodity  $q \in Q$  assigned to path  $p \in P(q)$ .

Given a partition  $\Xi_1 \cup \dots \cup \Xi_K$  of the uncertainty set  $\Xi$ , the  $k$ -adaptive routing scheme restricts the routing functions to piece-wise constant functions defined as  $y_{qp}(\xi) = y_{qp}^k \Leftrightarrow \xi \in \Xi_k$ . The formulation follows, where  $Y \equiv \{0, 1\}$  for nonbifurcated flows and  $Y \equiv [0, 1] \subset \mathbb{R}$  for bifurcated flows:

$$\begin{aligned}
(kRNL) \quad & \min \sum_{ij \in E} c_{ij} x_{ij} \\
& \text{s.t.} \\
& \sum_{q \in Q} \sum_{p \in P(q)} (\bar{d}^q + \hat{d}^q \xi^q) \delta_{qp}^{ij} y_{qp}^k \leq x_{ij} \quad \forall ij \in E, \forall \xi \in \Xi_k, \quad (1) \\
& k \in \{1, \dots, K\} \\
& \sum_{p \in P(q)} y_{qp}^k = 1 \quad \forall q \in Q, k \in \{1, \dots, K\} \\
& y_{qp}^k \in Y, x_{ij} \geq 0, x_{ij} \in \mathbb{Z} \quad \forall q \in Q, \forall p \in P(q), \\
& \quad \quad \quad \forall ij \in E, \\
& \quad \quad \quad k \in \{1, \dots, K\}
\end{aligned}$$

### 3 Solution strategy

**Dynamic Partitioning** In [3], the authors identify that there is a set of active uncertain parameters  $\hat{\xi}$  that restricts the objective function. The active uncertain parameters are the values of the uncertain parameters that corre-

respond to the constraints with minimum slacks. In  $kRNL$  the only constraints that involve uncertainty are constraints (1). We can define the set  $\hat{\Xi}_k$ , of the active uncertain parameters for a set  $\Xi_k$  of the partition as follows. Given a solution  $(\tilde{x}_{ij}, \tilde{y}_{qp}^k)$  we define  $\hat{\Xi}_k = \{\hat{\xi} \mid \hat{\xi} = \arg \min_{\xi \in \Xi_k} (\tilde{x}_{ij} - \sum_{q \in Q} \sum_{p \in P(q)} (\bar{d}^q + \hat{d}^q \xi^q) \delta_{qp}^{ij} \tilde{y}_{qp}^k), \forall ij \in E\}$ , where only one element will be selected for each constraint (1). This shows that we must partition the uncertainty set in such a way as to guarantee that the uncertain parameters for the active constraints do not all lie in one set of the partition. The authors use Voronoi diagrams to further partition each set  $\Xi_k$  of the current partition as follows. They impose that each element selected of  $\hat{\Xi}_k$  belongs to a single set of the new partition. Hence, their approach creates a partition tree, called nested partitioning, where the children of an element represent the sets partitioning the element.

**Bounds** We implement two bounds to improve computational tractability. The first bound provides an estimate of the gap between the  $k$ -adaptive solution and a dynamic solution. The set of all active uncertain parameters generated along the iterations are collected and used to obtain a lower bound on the dynamic solution. A useful property of this bound is that as we obtain more samples the lower bound improves. The second bound is an upper bound on the objective value of a subsequent iteration of the algorithm. The objective values using the nested approach do decrease monotonically through iterations, and a solution to a  $k$ -partition  $(x_{ij}, y_{qp}^k)$  at one iteration provides feasible solution for their children partitions at the next iteration.

**Benders Decomposition** Dynamic partitioning suffers from an inherent characteristic of multiplying the number of variables and constraints after each iteration. This impacts tractability of problem resolution, so that decomposition techniques are introduced in order to minimize the impact of this characteristic. For the bifurcated case, we take advantage of the natural staircase block structure of the  $kRNL$  formulation, where each block is related to a set of the partition, to implement a Benders based decomposition. We use a branch-and-cut approach and break the problem into one that designs edge capacities (master problem defining variables  $x_{ij}, ij \in E$ ) and a sequence of other problems that checks feasibility of the designed edge capacities with respect to the uncertain demand requirements.

Specifically, given an integer solution  $\tilde{x}_{ij}$  for the master problem (typically obtained at an integer node of the branch-and-bound tree of the master

problem), we solve a feasibility subproblem for each element of the partition  $k \in K$ . Notice that each subproblem is the following robust linear program

$$\begin{aligned}
\min \quad & \sum_{ij \in E} s_{ij} \\
\text{s.t.} \quad & \sum_{q \in Q} \sum_{p \in P(q)} d_q(\xi) y_{qp}^k \delta_{ij}^p - s_{ij} \leq \tilde{x}_{ij}, \\
& \forall ij \in E, \forall \xi \in \Xi_k \\
& \sum_{p \in P(q)} y_{qp}^k = 1, \forall q \in Q \\
& s_{ij} \geq 0,
\end{aligned}$$

where variables  $s_{ij}$  are additional slack variables. If the optimal solution cost of subproblem  $k$  is positive, we use the dual optimal solution  $(\tilde{\pi}_{ij}, \tilde{\mu}_q)$  to add a strengthened Benders cut to the master problem:

$$\sum_{ij \in E} \lceil \frac{\tilde{\pi}_{ij}}{m} \rceil x_{ij} \leq \lceil \frac{\sum_{q \in Q} -\tilde{\mu}_q}{m} \rceil,$$

where  $m = \min_{ij \in E} \tilde{\pi}_{ij}$ . Non zero values of slack variables complement and round up master problem solutions and are used as upper bounds for next iteration of master problem. We finally improve our algorithm by using robust cut set inequalities (e.g. [5],[8]).

## 4 Implementation and Results

The purpose of our experiment is two-fold:

- For nonbifurcated flows, we show the cost reductions provided by  $k$ -adaptive routing scheme over static. This is the first attempt to improve static solutions as the integrality of the second stage variables prevents us from enumerating the extreme points of  $\Xi$  (see [14]) or using classical decomposition algorithms (see [7]).
- For bifurcated flows, we compare the solution times and costs of  $k$ -adaptive routing scheme with those of static and volume routings.

**Instances** Network instances (6) available from SNDlib [10] were utilized with the characteristics given in Table 1a. The uncertainty set is given by

$\Xi = \{\xi \mid \xi \in [0, 1]^{|Q|}, \sum_{q \in Q} \xi^q \leq \Gamma\}$  and we define two  $\Gamma$ s based on the probabilistic bound of 25% and 1% (e.g. [4],[8]).

**Algorithms Specification** Algorithms were coded in Julia [6] using JuMP and JuMPeR packages and Gurobi 6.1. All algorithms were run in an Intel CORE i7 CPU 3770 machine. A limit of 3600 seconds of computing time was given for each instance and iteration. We run all algorithms according to configuration Table 1c. The different methods were used to solve the equivalent robust static problem at each iteration, according to formulation *kRNL*. Original formulation is formulation *kRNL* solved using JuMPeR package from Julia by cutting plane method. For Decomposition method, Benders primal subproblems were solved through robust deterministic reformulation set using JuMP lazy callback functions. The subproblem are solved through the the dual simplex method to leverage the fact that only the right hand side ( $\tilde{x}_{ij}$ ) of constraints change.

**Routing schemes** We test bifurcated and nonbifurcated flows routing schemes according to configuration Table 1b. The volume routing scheme, a variant of affine routing and only valid for bifurcated flows, was implemented based on [8], where each path variable was defined as  $y_{qp}^k = y_{qp}^{0k} + y_{qp}^{1k}d^q$ . Full partitioning means *k*-adaptive routing scheme where at the end of each iteration we add active uncertain parameters for all capacity constraints (1) of our formulation. Partial partitioning means *k*-adaptive routing scheme where (i) we only add active uncertain parameters referent to the set of the partition that is restricting the objective value and (ii) we restrict to 10 active uncertain parameters referent to the constraints of this set with minimum slack. We do this as a trade off between computational tractability and quality of the solution. Static routing scheme solution is provided as a result of first iteration of our *k*-adaptive solution. We have predetermined a maximum of 7 paths for each commodity (if they exist).

**Results** We present here a subset of our results. Table 2 compares solutions provided by different bifurcated and nonbifurcated flows routing schemes. There are a number of instances that do not achieve an optimal solution under the time limit, marked as *M*. Due to our strategy to reduce number of possible paths, volume routing provides same results as static routing. For bifurcated flows the results for our instances show that the *k*-adaptive routing scheme can provide good intermediary solution between an affine (volume) and dynamic routing. The cost reduction is higher when the protection level



Instance	V	E	Q	$\Gamma^{0.25}$	$\Gamma^{0.01}$	Routing Scheme	Code	Method
abilene	12	15	65	6	19	Static	C	Original formulation
polska	12	18	66	6	19	Volume (Affine)	C+C	Original formulation + Cut Set
pdh	11	34	24	4	12	Full Partitioning	D	Decomposition
di-yuan	11	42	22	4	11	Partial Partitioning	D+C	Decomposition + Cut Set
nobel-us	14	21	91	7	23			
atlanta	15	22	105	7	24			

(a) Instances Profile

(b) Routing Schemes

(c) Method Configuration

Table 1  
Problem Set Up

Instance	$\Gamma$	Static BF	Volume BF*	Full partitioning BF*	Partial partitioning BF*	Static NBF	Full partitioning NBF*	Partial partitioning NBF*
abilene	6	3.13e+00	0	0.63	0.63	3.22e+00	2.08	2.08
	19	3.69e+00	0	0.54	0.54	3.69e+00	0	0
polska	6	2.61e+02	0	4.12	3.10	2.74e+02	M	M
	19	2.95e+02	0	1.02	0.97	3.02e+02	M	M
pdh	4	8.51e+05	0	0	0	8.69e+05	0	0
	12	8.60e+05	0	0	0	8.69e+05	0	0
di-yuan	4	5.24e+06	0	2.10	1.12	5.39e+06	0.20	0.20
	11	5.37e+06	0	0	0	5.51e+06	0	0
nobel-us	7	2.94e+05	0	2.47	2.30	3.05e+05	M	M
	23	3.23e+05	0	0.61	0.61	3.26e+05	M	M
atlanta	7	2.00e+05	0	2.01	1.50	2.01e+05	1.50	1.50
	24	2.15e+05	0	0.93	0.93	2.17e+05	M	M

\*Results as percentage of reduction:  $100\% \cdot (\text{Static-}^*)/\text{Static}$ Table 2  
Bifurcated (BF) and Non Bifurcated (NBF) Flows Solution

is higher (25%). Partial partitioning was able to give results comparable to full partitioning.

Table 3 compares time performance between different methods for bifurcated routing schemes. We have limited our algorithms to 3 iterations. Larger instances are more difficult to solve, specially as the problem grows between iterations, and decomposition could provide a viable alternative to improve performance in some instances .

## 5 Conclusions

We have performed a numerical experiment in a search to provide efficient routings and less conservative solutions to RNL. The  $k$ -adaptive routing scheme was able to provide improved solutions when compared to affine (volume) and static solutions. The method utilized does suffer of dimensionality issues so that special techniques to maintain tractability are fundamental. In fact, the

		Routing Scheme and Method Code					
		Static		Partial partitioning			
Instance	$\Gamma$	C	C+C	C	D	C+C	D+C
abilene	6	1.23	1.58	20.30	44.86	20.26	17.09
	19	3.78	1.83	134.47	103.28	43.56	31.76
polska	6	70.36	59.34	M	M	2422.96	1024.28
	19	27.50	24.92	M	M	3505.23	1823.27
pdh	4	5.27	3.65	104.58	136.53	69.57	26.65
	12	2.14	1.14	32.05	105.29	28.87	44.33
di-yuan	4	1.52	1.40	835.09	283.57	980.32	83.68
	11	0.90	0.63	38.84	88.60	15.88	23.92
nobel-us	7	60.78	55.21	624.89	M	476.18	1978.23
	23	31.27	31.46	M	M	1563.33	2053.49
atlanta	7	20.32	14.89	68.10	93.27	31.85	32.24
	24	163.70	138.33	500.27	845.23	458.44	248.84

Table 3  
Bifurcated Flows Time Performance (s)

$k$ -adaptive partial partitioning can provide good results, when compared to full partitioning and have better time performance. Our preliminary results also showed that Benders decomposition can be efficient to speed up instances.

## References

- [1] Walid Ben-Ameur and Mateusz Zotkiewicz. Robust routing and optimal partitioning of a traffic demand polytope. *ITOR*, 18(3):307–333, 2011.
- [2] D. Bertsimas and C. Caramanis. Finite Adaptability in Multistage Linear Optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, Dec 2010.
- [3] Dimitris Bertsimas and Iain Dunning. Multistage Robust Mixed-Integer Optimization with Adaptive Partitions. *Oper Res*, 64(4):980–998, 2016.
- [4] Dimitris Bertsimas and Melvyn Sim. The Price of Robustness. *Oper. Res.*, 52(1):35–53, January 2004.
- [5] Arie M. C. A. Koster, Manuel Kutschka, and Christian Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.
- [6] Miles Lubin and Iain Dunning. Computing in Operations Research using Julia. *CoRR*, abs/1312.1431, 2013.
- [7] Sara Mattia. The robust network loading problem with dynamic routing. *Comp. Opt. and Appl.*, 54(3):619–643, 2013.

- [8] Sara Mattia and Michael Poss. A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. *Submitted, Available at [www.optimization-online.org](http://www.optimization-online.org)*, 2014.
- [9] Michel Minoux. Robust network optimization under polyhedral demand uncertainty is NP-hard. *Discrete Applied Mathematics*, 158(5):597–603, 2010.
- [10] Sebastian Orłowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. SNDlib 1.0 - Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.
- [11] A. Ouorou and J.-P. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. *Proc 6th Int Workshop Design Reliable Commun Networks*, pages 1–4, 2007.
- [12] Adam Ouorou. Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty. *Computers & OR*, 40(1):318–327, 2013.
- [13] Michael Poss. A comparison of routing sets for robust network design. *Optimization Letters*, 8(5):1619–1635, 2014.
- [14] Michael Poss and Christian Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, 2013.
- [15] Krzysztof Postek and Dick den Hertog. Multistage Adjustable Robust Mixed-Integer Optimization via Iterative Splitting of the Uncertainty Set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.