Optimal routing configuration clustering through dynamic programming

Yacine Al Najjar, Stefano Paris, Jocelyne Elias, Jeremie Leguay, Walid Ben-Ameur

Programmation dynamique pour l'optimisation du clustering des configurations réseau

Yacine Al Najjar, Stefano Paris, Jocelyne Elias, Jérémie Leguay, et Walid Ben-Ameur[+]

Mathematical and Algorithmic Sciences Lab - Paris Research Center, Huawei Technologies France.
Samovar, UMR5157, CNRS, Telecom Sud Paris, France[+].

Les réseaux SDN (*software-defined networking*) permettent d'optimiser globalement la configuration des réseaux en fonction du trafic. Idéalement, le réseau devrait être reconfiguré de manière dynamique afin de permettre une meilleure utilisation des ressources. Néanmoins, en pratique, les changements de routes ne peuvent pas être trop fréquents à cause de la lente mise à jour des équipements et de la dynamicité individuelle des flux. Pour trouver un bon compromis entre fréquence de reconfiguration et l'efficacité de l'utilisation des ressources, une nouvelle approche d'ingénierie de trafic basée sur le *clustering robuste* a été proposée dans [SFC[+]18]. L'idée principale est de regrouper des scénarios de trafic futurs présentants des similitudes de routage et temporel. Pour résoudre le problème de clustering robuste d'une manière optimale et rapide, nous proposons un nouvel algorithme basé sur la *programmation dynamique*. Nous comparons l'algorithme avec l'approche heuristique de [SFC[+]18] et la résolution de l'ILP sur des instances réelles. Les résultats montrent que notre approche est efficace pour trouver la solution optimale et améliore les performances du routage par rapport à l'heuristique.

Mots-clefs : Réseaux logiciels, routage robuste, calcul de chemins, programmation dynamique.

1 Introduction

In recent years, the control paradigm of Software Defined Networking (SDN) [KRV[+]15], which was originally targeting data-center networks, has gained momentum up to encompassing worldwide interconnection systems. Carrier Operators have been building Wide Area Networks (WAN) managed by a centralized SDN controller to provide long-haul and cost effective connectivity.

The dynamic nature of IP traffic makes Traffic Engineering (TE) challenging. Ideally, the network should be dynamically reconfigured as traffic evolves to efficiently use network resources and to reduce operational costs. Unfortunately, reconfiguration cannot be too frequent due to route stability, control overhead and individual flows dynamics. The solutions that have been designed to cope with traffic and network state variations can be broadly classified into three main classes: dynamic TE, static TE, and semi-static TE. While different in the way they calculate network configurations, all techniques require the use of Traffic Matrices (TMs) periodically collected by a network monitoring tool. Dynamic TE, like [JKM[+]13], uses such information to predict the next system state and compute the corresponding optimal routing configuration using linear programming or fast approximation algorithms. The accuracy of the prediction highly affects the optimality of the computed solution while frequent network reconfigurations result in control plane congestion due to the low speed of flow programming in hardware. In contrast, static TE, such as oblivious routing [BAK03, ACF[+]06] (called also *stable* or *static* routing) monitors TMs over a long period of time and computes the TE configuration that minimizes the worst deviation with respect to the sequence of all optimal configurations. This class of TE policies keeps the network configuration stable, but it inevitably suffers from low optimality during most of the operational time. Semi-static TE approaches such as [BZ11] combine both static and dynamic TE to approximate the optimal sequence of configurations with a limited set of routing solutions computed over clusters of TMs. In [SFC[+]18] we have formalized

the Clustered Robust Routing (CRR) problem to pre-compute a set of routing configurations that improve the overall network utilization and guarantee a smooth transition during the reconfiguration. To limit the number of reconfigurations, we have formulated the clustering problem imposing a limit on the maximum number of clusters and a minimum cluster length, which corresponds to a minimum holding time of the configuration. In this paper, we formally prove that the CRR problem can be solved in polynomial time when the underlying robust routing problem admits splittable flows. Furthermore, we design an efficient algorithm based on dynamic programming to compute the optimal solution. Finally, we analyze and evaluate our algorithm on realistic network scenarios showing that it outperforms CPLEX even on small instances.

2 Clustered Robust Routing problem

In this section we briefly present the CRR problem [SFC+18] and introduce some key notations. Given a network topology and a sequence of traffic matrices (TMs), which might represent the evolution over time of end-to-end connections (i.e., demands), the CRR problem consists in splitting the sequence into smaller clusters of contiguous TMs with a single routing configuration applied to each cluster. Since a single routing configuration is applied to all TMs of the cluster, the goal is to find the clusters and corresponding routing configurations that minimize the worst deviation with respect to Dynamic-TE.

Fig. 1 presents an example of sequence of TMs composed of two demands (d_1 and d_2) and three clusters (C_1, C_2, and C_3). The three routing configurations (R_1, R_2, and R_3) associated to the three clusters are feasible for any TM within the cluster and minimize the deviation of the objective function with respect to the optimal routing configuration of each TM.

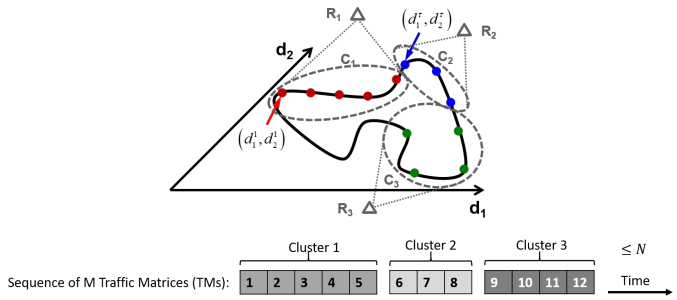


Fig. 1: Example of a possible clustering with $M = 12$, $N = 3$ and $L = 3$.

We represent the network infrastructure as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links. Each link $e \in \mathcal{L}$ has a capacity $c_e \geq 0$ and a cost $w_e \geq 0$ per flow unit. We assume that we are given the Traffic Matrix (TM) at each time slot τ in $\{1, ..., M\}$ by using a TM prediction module, and for each demand $h \in \mathcal{H}$ we have the amount of flow d_h^τ that must be routed from its origin to its destination at time slot τ. A *Routing Configuration (RC)* is both the *set of paths* that will be used to route demands from their origins to their destinations and *each demand percentage* that will flow through each path. The main goal is to find the RCs for these TMs such that we minimize the total cost over time. The more intuitive way to do that is to determine at each time slot the RC that minimizes the cost for the corresponding TM. However, this method cannot be implemented in practice for stability issues since it would cause frequent network reconfigurations. Therefore, we impose a minimum length L on the holding time of the RCs and a maximum number N of clusters (in this work, we consider one RC per cluster); an example is illustrated in Fig. 1. Reformulated in another way, we want to find a maximum number N of contiguous clusters of TMs with a minimum holding time of L that minimize the overall routing cost. This problem can be solved in an offline setting to pre-compute RCs. An online algorithm can then decide when to activate those RC. In this paper we will focus on the first problem.

3 Algorithmic approach

The proposed algorithm can be divided in two parts. In the first part (Sec. 3.1) we compute the value of the best robust routing (or determine the best RC) for each possible cluster and in a second part (Sec. 3.2) we use dynamic programming to find the best clustering of the TMs.

3.1 Robust routing for a cluster

The goal here is to find the best RC for a cluster including all TMs between time slots t_1 and t_2, such that we minimize the total cost over the cluster. We will give now a formal description of this problem using a linear program. Variables f_e^h are used to express the proportion of demand h routed over link e. Constraint (2) ensures that flow conservation is guaranteed at each node. Notice that $OUT(i)$ is here denoting the set of links going out of node i while $IN(i)$ denotes links oriented toward i. Constraint (3) guarantees that the total amount of flow using link e at time τ does not exceed its capacity.

$$[RR(t_1,t_2)] : \min \sum_{t_1 \leq \tau \leq t_2, e \in \mathcal{L}, h \in \mathcal{H}} w_e d_h^\tau f_e^h \tag{1}$$

$$\text{s.t.} \sum_{e \in OUT(i)} f_e^h - \sum_{e \in IN(i)} f_e^h = \begin{cases} 1 & \text{if } i = origin(h) \\ -1 & \text{if } i = destination(h) \\ 0 & otherwise \end{cases} \quad \forall i \in \mathcal{N}, \forall h \in \mathcal{H} \tag{2}$$

$$c_e \geq \sum_{h \in \mathcal{H}} d_h^\tau f_e^h \qquad \forall \tau \in \mathcal{T} : t_1 \leq \tau \leq t_2, \forall e \in \mathcal{L} \tag{3}$$

$$0 \leq f_e^h \leq 1 \qquad \forall e \in \mathcal{L}, \forall h \in \mathcal{H} \tag{4}$$

The formulation above is compact (i.e., it has a polynomial number of variables and constraints). When the network size becomes too large, other formulations based on either path or rooted tree variables might be used.

3.2 Dynamic programming algorithm

Let $V(t,c)$ be the optimal cost obtained when all TMs between 1 and t are considered and only c clusters are allowed. Since each cluster should contain at least L TMs, we can write that $V(t,c) = RR(1,t)$ for $1 \leq t \leq L$. When $c = 1$, we clearly have $V(t,1) = RR(1,t)$ for $1 \leq t \leq M$. For $L+1 \leq t \leq M$ and $1 \leq c \leq N-1$ the following recurrence relation can be used:

$$V(t,c+1) = \min_{L \leq \tau \leq t-1} V(t-\tau,c) + RR(t-\tau+1,t). \tag{5}$$

It simply expresses the fact that $V(t,c+1)$ is obtained by clustering the TM of slot t with some $\tau-1$ previous matrices and considering the optimal solution at time $t-\tau$ with c possible clusters. τ is of course larger than L and can not go above $t-1$. The value of τ minimizing the sum $V(t-\tau,c)+RR(t-\tau+1,t)$ is then used to get $V(t,c+1)$. The optimal solution of the robust clustering problem is obtained for $c = N$ and $t = M$. The algorithm requires the computation of $O(M^2)$ compact linear programs and $O(NM^2)$ operations, thus it runs in polynomial time.

Notice that due to the constraint related to the minimum size of a cluster, the number of clusters cannot be greater than or equal to $\lfloor M/L \rfloor$. This clearly implies that when $N \geq \lfloor M/L \rfloor$, the constraint related to the number of clusters is already implied by the holding-time constraint. Consequently, $V(M,N) = V(M,\lfloor M/L \rfloor)$ and one can still use the above recurrence to compute it. However, a faster algorithm can be used in this case by skipping the index c and considering the quantity $V(t)$ representing the optimal clustering for TMs between 1 and t satisfying the holding-time constraint. The recurrence relation becomes:

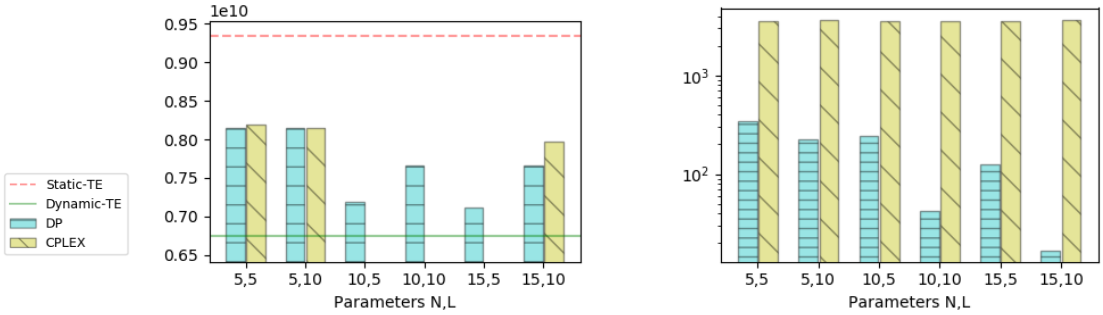
$$V(t) = \min_{L \leq \tau \leq \min(t-1,2L-1)} V(t-\tau) + RR(t-\tau+1,t). \tag{6}$$

4 Numerical Results

To evaluate our algorithm we generate a random network topology and a sequence of $M = 120$ traffic matrices. The network is composed of 11 nodes and 32 links, whereas each traffic matrix has 24 demands. The aggregated traffic of all demands is generated as a sinusoidal wave to obtain a diurnal and nocturnal pattern. Traffic is then distributed across demands according to a Pareto distribution with few large flows and several small flows. We compare our Dynamic Programming approach (label 'DP') against CPLEX executed on the MILP model described in [SFC+18]. We further analyze Dynamic-TE and Static-TE which represent the lower and upper bounds, respectively. Dynamic-TE computes a routing configuration for each TM using a column generation approach, whereas Static-TE finds the unique routing configuration that minimizes the deviation of the objective function for all TMs using a column and row generation algorithm. Static-TE can be considered as a special case of CRR with a single cluster with all TMs (i.e., $N = 1$ and $L = M$). We observe that Dynamic-TE generates M clusters with a single TM each, while Static-TE computes only a single cluster with all TMs. In all our simulations, CPLEX is configured with one hour time limit and an optimality gap of 1%.

Fig. 2a depicts the value of the objective function of the algorithms as a function of the parameters N and L (with $N = \{5, 10, 15\}$ and $L = \{5, 10\}$) while Fig. 2b shows their execution time on a logarithmic scale.

We can observe that, while the network instance might seem rather small, CPLEX never finds the optimal solution of the original CRR problem. More specifically, for three out of six cases ($(10,5)$, $(10,10)$ and $(15,5)$) no feasible solution has been found within 1 hour, while for all the other cases the computed clustering provides a worse value than our DP approach. In contrast, our approach always finds the optimal solution in less than 5 minutes.



(a) Objective Function

(b) Execution Time (s)

References

[ACF+06] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Raacke. Optimal oblivious routing in polynomial time. *ACM Symp. on Theory of Computing*, 2006.

[BAK03] Walid Ben-Ameur and Hervé Kerivin. New economical virtual private networks. *Communications of the ACM*, 46 (6):69–73, 2003.

[BZ11] Walid Ben-Ameur and Mateusz Zotkiewicz. Robust routing and optimal partitioning of a traffic demand polytope. *ITOR*, 18(3):307–333, 2011.

[JKM+13] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined wan. *SIGCOMM*, 2013.

[KRV+15] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proc. IEEE Journal*, 2015.

[PR13] Michael Poss and Christian Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61:180–198, 03 2013.

[SFC+18] Davide Sanvito, Ilario Filippini, Antonio Capone, Stefano Paris, and Jeremie Leguay. Adaptive robust traffic engineering in software defined networks. In *Proc. IFIP Networking*, 2018.

[TKB+07] V. Tabatabaee, A. Kashyap, B. Bhattacharjee, R. J. La, and M. A. Shayman. Robust routing with unknown traffic matrices. *Proc. IEEE Journal*, 2007.

[WXQ+06] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. Cope: Traffic engineering in dynamic networks. *SIGCOMM*, 2006.