



**HAL**  
open science

## La ROUTOURNE va tourner

Quentin Bramas, Jean-Romain Luttringer, Pascal Mérindol

► **To cite this version:**

Quentin Bramas, Jean-Romain Luttringer, Pascal Mérindol. La ROUTOURNE va tourner : le retour des tours de détours sous leurs plus beaux atours. AlgoTel 2024 – 26èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2024, Saint-Briac-sur-Mer, France. hal-04567055v1

**HAL Id: hal-04567055**

**<https://hal.science/hal-04567055v1>**

Submitted on 5 May 2024 (v1), last revised 13 May 2024 (v2)

**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.

# *La ROUTOURNE va tourner : le retour des tours de détours sous leurs plus beaux atours*

Quentin Bramas<sup>1</sup> et Jean-Romain Luttringer<sup>1</sup> et Pascal Mérindol<sup>1</sup>

<sup>1</sup> *Université de Strasbourg, Laboratoire ICube*

---

Le routage par segments (SR) offre un contrôle précis sur les chemins empruntés : il spécifie dans les paquets IP une liste de détours, appelés *segments*. Le nombre de détours pouvant être spécifiés est cependant limité par le matériel. Lors du calcul de listes de segments, il est donc nécessaire d'en limiter la taille. Bien que des solutions aient été proposées pour calculer ces listes, celles-ci manquent de généralité, et ne sont pas toujours optimales ou efficaces. Nous présentons ROUTOURNE, une méthode permettant de détourner les algorithmes de routage afin que ces derniers calculent, non pas simplement un chemin physique optimal à traduire en liste de segments a posteriori (sans garantie sur sa taille), mais directement les listes de segments optimales et déployables par le matériel sous-jacent. ROUTOURNE facilite ainsi le déploiement de stratégies d'ingénierie de trafic et de politiques avancées, notamment pour équilibrer la charge depuis les sources. Malgré une route truffée de défis surprenants – en particulier, la perte d'isotonie induite par SR, ROUTOURNE s'avère efficace, induisant au pire un surcoût linéaire. Son exactitude et optimalité ont été prouvées, et son efficacité évaluée en la généralisant à plusieurs algorithmes de calcul de chemins plus ou moins complexes.

**Mots-clefs :** Calcul de chemins, Segment Routing, Ingénierie de Trafic, Qualité de Service

---

## 1 Introduction

Les réseaux IP utilisent le routage "best-effort" au saut par saut, où les chemins minimisent le *coût IGP*, une métrique additive définie par l'opérateur. Cependant, il est fréquent de devoir s'affranchir de ces chemins pour leur privilégier ceux offrant des garanties concrètes, comme une latence bornée pour certains flux premium ou le contournement d'équipements défectueux.

Le calcul de tels chemins est un défi en soi, surtout lorsque plusieurs métriques sont prises en compte simultanément. Cependant, nous nous penchons ici sur l'étape suivante, souvent négligée : le déploiement de ces chemins contraints en détournant le trafic des chemins "best-effort" utilisés dans le réseau.

En pratique, ces détours sont construits à l'aide du routage à la source relâché, dont l'implémentation la plus répandue est Segment Routing (SR). SR permet de spécifier directement dans le paquet une liste de détours. Le paquet va de détour en détour en suivant les plus courts chemins IGP entre ces derniers. Cependant, le nombre de détours (appelés "segments") spécifiés est limité par le matériel sous-jacent. Cette contrainte, appelée Profondeur Maximale des Segments (PMS), implique que certains chemins physiques peuvent ne pas être déployables (car nécessitant trop de segments pour être décrits). Cependant, prendre en compte cette contrainte lors du calcul des chemins pose plusieurs défis qui ont des conséquences inattendues, comme la perte d'optimalité des sous-chemins (ou isotonie), et rendent impossible l'utilisation naïve des algorithmes existants. Il est donc nécessaire de fournir un cadre correct et efficace pour intégrer SR sans compromettre les performances des calculs.

Nous présentons ROUTOURNE, une méthode facilitant le déploiement d'ingénierie de trafic avancée. ROUTOURNE permet d'adapter des algorithmes de calculs de chemins existants afin que ces derniers retournent des listes de segments optimales et déployables. ROUTOURNE repose sur deux éléments clés :

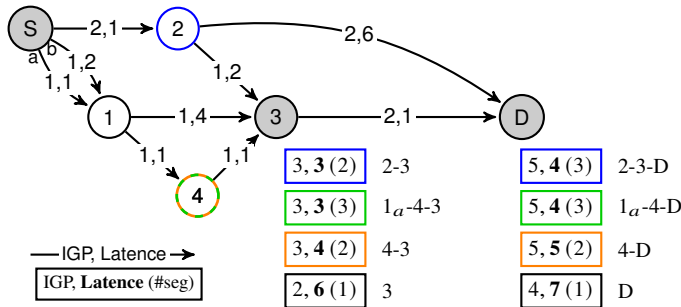
- La traduction des chemins explorés en listes de segments minimales *pendant l'exploration du graphe afin de guider la recherche* ;
- La modification de la fonction de comparaison des chemins afin de considérer le nombre de segments correctement, malgré la perte de l'isotonie standard.

L’optimalité et l’exactitude de  $\text{ROUTOURN}^E$  ont été démontrées et son efficacité évaluée sur des réseaux variés, exhibant son faible surcoût linéaire. Le code de  $\text{ROUTOURN}^E$  et des expériences est public <sup>†</sup>.

## 2 Défis et tour de l’état de l’art

L’approche type pour obtenir une ou des listes de segments est d’*encoder* un unique chemin donné en entrée en un nombre minimal de segments [GDLT16]. Bien qu’utile pour certains scénarios d’usage, ce modèle de traduction a posteriori présente des limites. Il ne tire pas parti de l’opportunité d’équilibrer la charge sur des chemins équivalents (e.g., offrant la même latence) et, plus problématique, il est dissocié du calcul du chemin. Il n’y a donc aucune garantie que le chemin passé en entrée soit encodable en moins de PMS segments, comme l’illustre la figure 1. Encoder le chemin de plus faible latence de S vers D résulte en une liste de segments de taille 3, qui pourrait ne pas être déployable sur du matériel fortement contraint. De ces observations émerge le premier défi : **intégrer directement l’algorithme d’encodage dans le calcul de chemins afin d’en guider la recherche en tirant partie de l’équilibrage de charge.**

Supposons celui-ci résolu, avec la connaissance du surcoût opérationnel permettant de déployer les distances explorées. Plusieurs métriques sont ainsi à considérer : non seulement les métriques du problème originel, mais aussi le nombre de segments nécessaires. Le calcul de telles distances multi-métriques partage le même principe que son homologue mono-métrique : chaque distance optimale est étendue afin de déterminer les meilleures distances vers des noeuds plus éloignés. La définition d’*isotonie* est cependant généralisée afin d’inclure l’ensemble des distances non-dominées, i.e., le *front de Pareto* des distances <sup>‡</sup>. Intuitivement, modifier le calcul afin d’étendre toutes les distances non-dominées (nombre de segments compris) semble suffisant pour garantir le calcul correct des listes de segments déployables.



**FIGURE 1** : Segments nécessaires pour encoder les distances optimales vers 3 et D.  $\text{ROUTOURN}^E$  résout le défi de l’isotonie perdue.

effet, le détour 3 doit être explicitement spécifié, au risque de laisser un paquet utiliser le lien  $(2, D)$ , dont la latence proscrit l’utilisation pour garantir la distance  $(5, 4)$ . En revanche, les distances orange et verte peuvent être étendues jusqu’à D sans nécessiter de segment supplémentaire. *Repousser* leur dernier détour (noeud 3) d’un saut (jusqu’à D) est suffisant pour garantir que les chemins utilisés possèdent au pire une distance de  $(5, 4)$ . Ainsi, la distance verte, originellement dominée par la distance bleue, devient une solution de qualité égale. Plus intéressant encore, la distance orange devient la seule distance encodable en 2 segments offrant une latence proche de l’optimale. Cette distance, pouvant être la seule déployable sur du matériel contraint, ne sera trouvée qu’en étendant une distance sous-optimale dominée.

Cette perte d’isotonie pose problème : si l’ensemble des distances non-optimales devaient être étendues, la complexité du calcul pourrait croître exponentiellement. Certaines méthodes de calcul de listes de segments contournent ce problème en utilisant un graphe transformé où la métrique du nombre de segments est isotone, ce qui permet l’utilisation de méthodes de calcul de chemins classiques [Aub20, LAM<sup>+</sup>22, LBN<sup>+</sup>15]. Cependant, le graphe transformé est une clique, et cette densité maximale génère un surcoût non

Pourtant, cela ne garantit pas de découvrir toutes les solutions optimales, tel qu’illustré en Figure 1. Supposons que l’on recherche le chemin de plus faible latence. Plusieurs distances mènent au noeud 3. La distance en bleue  $(3, 3(2))$  domine les distances verte et orange; ces dernières possédant des poids égaux ou pires. Selon le principe d’optimalité des sous-chemins, seule la distance bleue serait alors étendue. Cependant, une fois la distance bleue étendue, un segment supplémentaire est nécessaire pour encoder celle-ci  $(5, 4(3))$  jusqu’au noeud D. En

<sup>†</sup>. <https://zenodo.org/records/10202365>

<sup>‡</sup>. Distances offrant un compromis intéressant, pour lesquelles il n’existe aucune distance meilleure sur l’intégralité des métriques.

négligeable pour la plupart des algorithmes. Il en résulte un second défi : **garantir que les meilleures listes de segments déployables seront correctement et efficacement calculées.**

### 3 ROUTOURNE : encoder et comparer à tour de bras

Afin de calculer des listes de segments optimales déployables, *ROUTOURNE* repose sur deux ingrédients : un algorithme d’encodage intégré et efficace ainsi qu’une modification de la comparaison des chemins afin de regagner l’isotonie perdue. Pour rester concis, nous présentons *ROUTOURNE* informellement en simplifiant certains principes. L’intégralité du cadre formel est disponible dans le rapport technique [BLM23].

**Encoder les distances.** Nous proposons un algorithme glouton pour l’encodage des distances à la volée. Celui-ci vérifie, selon les besoins, les propriétés spécifiées comme par exemple contourner un élément en panne. À chaque extension de distance, notre algorithme évalue si la nouvelle distance découverte est encodable en un unique segment (i.e., en repoussant le dernier détour spécifié d’un saut supplémentaire). Il est alors nécessaire de s’assurer que les chemins sous-jacents au segment utilisé vérifient les propriétés désirées, et possèdent des distances égales ou meilleures à la distance considérée. Dès lors que l’extension par une nouvelle arête rend la distance non-encodable par un unique segment, un segment (i.e., détour) supplémentaire est ajouté. Lors de la prochaine extension, l’algorithme d’encodage suit le même principe, mais considère ce détour intermédiaire comme nouvelle source. Une organisation efficace de la base de données des segments permet de réaliser cette vérification pour un coût négligeable.

Bien que l’encodage soit guidé par le chemin en cours d’exploration par l’algorithme, notre méthode autorise des *écarts* si ces derniers satisfont les propriétés et distances du chemin considéré. Ainsi notre encodage permet de tirer parti de l’équilibrage de charge sur des chemins au moins aussi *bons que le guide*.

**Redéfinition de la non-dominance.** Comme expliqué et illustré dans les parties précédentes, étendre uniquement les distances non-dominées est insuffisant pour garantir la découverte de toutes les distances optimales avec SR. Considérer l’ensemble des distances comme potentiellement optimales et toutes les étendre n’est pas une option réaliste en terme d’efficacité. Dans *ROUTOURNE*, nous révisons la définition de l’ensemble des distances à étendre pour assurer la découverte efficace des solutions optimales.

Les métriques standards telles que le coût IGP ou la latence restent isotones, seul le nombre de segments se comporte comme une métrique inhabituelle. *ROUTOURNE* analyse donc les cas dans lesquels des distances dominées pourraient nécessiter un nombre de segments égal à, ou meilleur que leurs concurrentes après extension, i.e., qui ne requerront pas de segment supplémentaire une fois étendues.

Le première observation essentielle est que le motif décrit ci-dessus, dans lequel l’écart en nombre de segments nécessaires pour une distance est réduit d’un segment par rapport à ses concurrentes, ne peut pas se produire consécutivement<sup>§</sup>. Une distance dominée nécessitant deux segments (ou plus) que ses concurrentes ne pourra donc jamais devenir optimale, et peut être supprimée de l’exploration sans risque.

La seconde observation est que des listes de segments partageant le même dernier détour évolueront de la même manière. En effet, ces listes construiront leurs détours successifs à partir des plus courts chemins IGP depuis une source intermédiaire commune. Par conséquent, il n’y a aucune raison qu’une de ces listes en particulier requière un segment supplémentaire et pas les autres. Ainsi, les distances dont les listes de segments partagent le même dernier détour peuvent être comparées selon les règles de dominance habituelles.

Ces deux observations nous permettent de conclure qu’une distance sous-optimale peut devenir optimale après extension uniquement si elle nécessite au pire un segment de plus que les distances qui la dominent, et ce alors que son dernier segment a une source différente de celles des listes qui la dominent. Seules ces distances additionnelles doivent être étendues, même si elles sont dominées au sens usuel du terme.

Cette relation de dominance *étendue* est à nouveau isotone, mais augmente le nombre de distances à considérer d’un facteur  $|V|$  au pire,  $V$  étant l’ensemble des noeuds du graphe (au pire, une distance dominée supplémentaire par source intermédiaire doit être maintenue). En pratique, ce surcoût dépasse rarement un facteur de 3 sur des graphes réalistes. Enfin, cette relation peut être ajustée selon plusieurs modèles pour

---

§. La preuve se déduit de l’optimalité des sous-chemins. Intuitivement, une listes de segments  $L'$  ayant été récemment augmenté d’un segment explore des chemins inclus dans les chemins explorés par une liste dont le détour est en amont. Par inclusion, si  $L'$  requiert un segment supplémentaire pour garantir une certaine distance,  $L$  également.

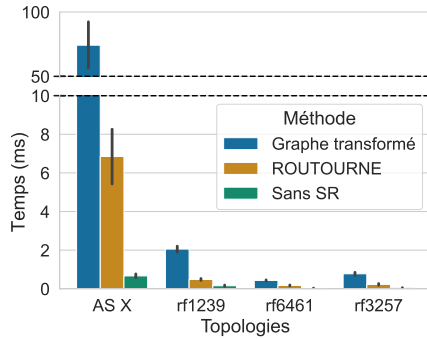


FIGURE 2 : Évaluation sur topologies réalistes de 2000, 1000, 370 et 300 arêtes respectivement.

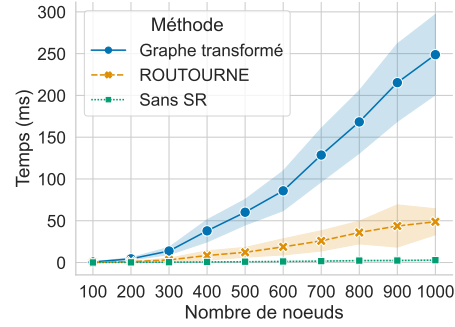


FIGURE 3 : Évaluation du temps de calcul sur des graphes synthétiques et denses.

obtenir une diversité maximale de listes de segments si l’opérateur souhaite bénéficier d’un large équilibrage de charge à la source (notre rapport technique [BLM23] donne davantage d’exemples).

**Détourner l’algorithme** En résumé,  $\text{ROUTOURNE}^E$  modifie l’algorithme originel afin de calculer efficacement des listes de segments optimales et déployables car (i) les distances explorées sont directement encodées en listes de segment à chaque extension (et non a posteriori), et (ii) seules les distances additionnelles vérifiant notre relation de dominance étendue sont maintenues et actualisées.

## 4 Évaluation : ROUTOURNE tourne plutôt bien

Les garanties théoriques de  $\text{ROUTOURNE}^E$  étant formellement démontrées [BLM23], cette évaluation se concentre sur la question de l’efficacité : *Quel est l’impact des modifications apportées par ROUTOURNE sur les performances de l’algorithme originel ?* Nous adaptions un algorithme multi-métrique, SAMCRA, de deux manières différentes : en utilisant  $\text{ROUTOURNE}^E$  et en lançant SAMCRA sur un graphe transformé (une clique), où le nombre de segments est nativement isotone. Notre analyse se concentre sur un cas difficile : le calcul des plus court chemins IGP respectant une borne supérieure sur la latence. Nous menons cette étude sur des graphes réalistes et synthétiques denses, afin de dégager d’éventuelles tendances.

Les résultats, présentés dans les figures 2 et 3, montrent que bien que  $\text{ROUTOURNE}^E$  augmente le temps d’exécution de l’algorithme (ce qui est attendu, puisque le problème devient multi-métrique), il reste nettement plus efficace que l’approche basée sur la transformation de graphe, notamment pour les réseaux de grande taille. Dans certains cas, le surcoût introduit par  $\text{ROUTOURNE}^E$  semble même négligeable.

## 5 Pour clôturer

$\text{ROUTOURNE}^E$  est une méthode générique augmentant les algorithmes de calcul de chemins afin qu’ils retournent des listes de segments optimales et déployables, facilitant ainsi l’ingénierie de trafic avancé. Elle repose sur deux éléments : un algorithme d’encodage efficace et relâché pour traduire les chemins en listes de segments, et une modification de la fonction de comparaison des distances. Cette redéfinition garantit la découverte efficace de solutions optimales malgré la perte d’optimalité des sous-chemins, fournissant diverses listes de segments immédiatement déployables selon l’équilibrage de charge désiré.

## Références

- [Aub20] Francois Aubry. *Models and Algorithms for Network Optimization with Segment Routing*. PhD thesis, UCLouvain, 2020.
- [BLM23] Quentin Bramas, Jean-Romain Luttringer, and Pascal Mérindol. A simple and general operational framework to deploy optimal routes with source routing, 2023.
- [GDLT16] Rabah Guedrez, Olivier Dugeon, Samer Lahoud, and Géraldine Texier. Label encoding algorithm for mpls segment routing. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, 2016.
- [LAM<sup>+</sup>22] Jean-Romain Luttringer, Thomas Alfroy, Pascal Mérindol, Quentin Bramas, François Clad, and Cristel Pelsser. Deploying near-optimal delay-constrained paths with segment routing in massive-scale networks. *Computer Networks*, 2022.
- [LBN<sup>+</sup>15] F. Lazzeri, Gianmarco Bruno, J. Nijhof, A. Giorgetti, and P. Castoldi. Efficient label encoding in segment-routing enabled optical networks. *2015 International Conference on Optical Network Design and Modeling (ONDM)*, pages 34–38, 2015.