



**HAL**  
open science

## Le Problème à trois Contraintes : Calcul et Déploiement de Segments de Routage

Jean-Romain Luttringer, Thomas Alfroy, Pascal Mérindol, François Clad,  
Cristel Pelsser

► **To cite this version:**

Jean-Romain Luttringer, Thomas Alfroy, Pascal Mérindol, François Clad, Cristel Pelsser. Le Problème à trois Contraintes : Calcul et Déploiement de Segments de Routage. ALGOTEL 2021 - 23èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2021, La Rochelle, France. hal-03216327

**HAL Id: hal-03216327**

**<https://hal.archives-ouvertes.fr/hal-03216327>**

Submitted on 3 May 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.

# Le Problème à trois Contraintes : Calcul et Déploiement de Segments de Routage

J-R. Luttringer<sup>1</sup>, T. Alfroy<sup>1</sup>, P. Mérindol<sup>1</sup>, F. Clad<sup>2</sup> et C. Pelsser<sup>1</sup>

<sup>1</sup> Université de Strasbourg, ICube <sup>2</sup> Cisco Systems

---

Longtemps freinée par des technologies peu extensibles et difficiles à automatiser, l'ingénierie de trafic retrouve peu à peu de son allant. D'une part, les services de communication émergents, comme le cloud gaming et l'industrie 4.0, nécessitent des chemins spécifiques offrant des garanties strictes. D'autre part, Segment Routing (SR), une technologie de routage par la source plus extensible que le plan de contrôle MPLS, offre aux opérateurs la possibilité de déployer des chemins contraints à grande échelle. Ces chemins peuvent par exemple respecter une contrainte de latence maximum tout en minimisant le "coût interne" pour l'opérateur (coût IGP). En effet, ce type de chemins est requis pour les applications nécessitant un haut niveau d'interactivité sans négliger la bande passante. Cependant, calculer de telles routes multi-contraintes est un problème NP-Difficile bien connu : DCLC. Bien que de nombreuses solutions existent, elles ne sont pas adaptées à Segment Routing qui ajoute une contrainte opérationnelle aux deux contraintes de qualité de service. De plus, ces propositions n'offrent généralement pas de garanties fortes en terme de temps d'exécution.

Dans ce travail, afin de proposer une solution exacte mais pratique et efficace, nous tirons parti des avantages et inconvénients de SR ainsi que des limites inhérentes aux réseaux d'opérateurs. Notre algorithme, BEST2COP, conçu pour être massivement parallélisable, résout efficacement DCLC même lorsque la double valuation du graphe est aléatoire. Que ce soit sur des graphes aux structures réelles ou aléatoires, BEST2COP résout DCLC en largement moins d'une seconde sur des domaines SR de plus de mille nœuds.

**Mots-clefs :** Multi-critères, DCLC, Ingénierie de Trafic, Calcul de Chemins, Segment Routing

---

## 1 Introduction

Certains services de communication exigent des garanties fortes sur plusieurs métriques, notamment le délai de propagation. Bien que la minimisation du *coût IGP* soit essentielle pour satisfaire les intentions de l'opérateur – en prenant en compte les ressources consommées comme la bande passante relative, il peut également être nécessaire de garantir un délai maximal. Cependant, calculer un chemin respectant une contrainte sur une première métrique additive tout en en minimisant une seconde est un problème NP-Difficile connu sous le nom de DCLC, *Delay Constrained, Least Cost*. Résoudre DCLC requiert de visiter l'intégralité des distances non-dominées, i.e., le front de Pareto [Deb05], pouvant croître de manière exponentielle avec le nombre de nœuds.

À cette difficulté inhérente au problème s'ajoute une difficulté technique : le déploiement des chemins calculés. Les opérateurs peuvent maintenant utiliser SR, *Segment Routing* [FNP<sup>+</sup>15], une technologie extensible de routage à la source. SR permet d'encoder la route *dans* le paquet sous forme d'instructions appelées *segments*. En pratique, le nombre de segments est contraint à MSD *Maximum Segment Depth*,  $\approx 10$ . À première vue, SR complexifie ainsi DCLC avec une dimension et une contrainte additionnelles. Cette métrique est ignorée par les solutions existantes qui ne prennent pas en considération la contrainte technique liée au déploiement des segments. De plus, elles n'offrent souvent pas de garanties sur leur temps d'exécution ou exactitude [GVRK18]. Nous allons voir comment cette limitation SR est en réalité un avantage dont nous tirons parti de plusieurs manières. De plus, des limites inhérentes à la mesure du délai de propagation nous permettent d'utiliser des structures efficaces pour la manipulation du front de Pareto. Notre algorithme, BEST2COP (Best Exact Segment Track for 2-Constrained Optimal Paths) [LAM<sup>+</sup>20], est aisément parallélisable et résout DCLC-SR, la version étendue de DCLC pour les domaines de routage SR, en largement moins d'une seconde, même sur de larges réseaux aléatoires et avec peu de threads.

## 2 DCLC-SR : Un Agneau Déguisé en Loup

Segment Routing permet d’encoder la route souhaitée en empilant des instructions (ou *segments*) dans le paquet. Chaque routeur sur le chemin dépile le segment actif et exécute l’instruction associée. Les deux principaux types de segments sont les *segments d’adjacence* et les *segments de nœud* : un segment d’adjacence stipule que le paquet doit être commuté via une interface spécifique (locale au routeur) alors qu’un segment de nœud indique plus “lâchement” que le paquet doit transiter via un certain nœud de routage. Un paquet contenant une telle instruction suit les plus courts chemins IGP vers ce nœud, en tenant compte d’ECMP. Ainsi, détourner un paquet des plus courts chemins IGP vers un nœud (e.g., via un chemin de latence moindre) demande l’utilisation d’un segment, dont le nombre total est limité à MSD.

Nous cherchons à résoudre DCLC dans un domaine SR, i.e., DCLC-SR. Ce problème considère trois métriques : le nombre de segments, le délai et le coût des chemins, notés respectivement  $M_0$ ,  $M_1$ ,  $M_2$ , sur lesquelles les contraintes  $c_0, c_1, c_2$  s’appliquent.

**Définition (DCLC-SR)** Résoudre DCLC-SR( $s, c_0, c_1$ ) consiste à calculer, pour une source  $s$  vers toutes les destinations du réseau, une liste de segments minimisant  $M_2$ , le coût IGP, tout en respectant les contraintes  $c_0$  et  $c_1$ , respectivement sur le nombre de segments et le délai de propagation  $\square$

DCLC-SR ajoute ainsi une dimension supplémentaire à un problème NP-difficile. Il convient donc de trouver un moyen efficace de résoudre DCLC tout en gérant cette nouvelle contrainte.

**La fidélité de la mesure** Résoudre DCLC requiert en théorie d’explorer un nombre potentiellement exponentiel de chemins. En effet, un chemin minimisant  $M_2$  peut, une fois étendu, ne plus respecter la contrainte sur le délai ( $c_1$ ) : il est donc nécessaire d’explorer et d’étendre également les chemins plus coûteux sur  $M_2$  mais meilleurs sur  $M_1$ . Ces chemins *non-dominés* forment le front de Pareto de la solution. Leur nombre pouvant être exponentiel, DCLC est intraitable dans les pires cas théoriques. Cependant, si l’une des deux métriques de qualité de service ( $M_1$  ou  $M_2$ ) possède un nombre fini de valeurs distinctes possibles, le nombre de distances non-dominées à stocker est borné : toutes les distances non-dominées peuvent ainsi être facilement stockées dans un tableau, indexées sur cette métrique bornée.

En pratique, il s’avère que  $M_1$  permet aisément de borner la taille du front de Pareto. En effet, la métrique délai est par construction bornée par une contrainte  $c_1$  stricte. De plus, bien que représentés avec une grande précision, les mesures réelles ont une fidélité, ou exactitude, beaucoup plus limitée à cause des défis techniques liés à la mesure du temps de propagation (e.g., synchronisation des instruments). Il est alors facile de discrétiser les distances  $M_1$  sans perte d’information pertinente. Par exemple, avec une fidélité de 0.1ms, le nombre maximal de distances sur le front de Pareto est de  $\Gamma = c_1 \times \frac{1}{0.1}$ . La fidélité avérée des délais de propagation mesurés permet donc de stocker l’intégralité du front de Pareto dans une structure efficace car pré-bornée, par exemple un tableau indexé sur  $M_1$  de taille  $\Gamma$ , et ce sans sacrifier l’exactitude de la solution.

**Le graphe SR multi-métrique** Bien que le grain de fidélité des mesures borne la complexité de DCLC, la contrainte ajoutée par SR (le nombre de segments) doit être correctement manipulée. Nous proposons pour cela une structure spécifique, le graphe SR Multi-Métrique, qui permet une exploration efficace de l’espace des solutions. Le graphe SR représente les segments sous forme d’arcs dans un graphe augmenté : les arcs entre les nœuds  $u$  et  $v$  représentent les segments pouvant être utilisés pour guider le paquet vers  $v$  depuis  $u$ . Sur notre construction [LAM<sup>+</sup>20], un arc peut représenter l’unique segment de nœud encodant les meilleurs chemins ECMP de  $u$  vers  $v$ . Dans ce cas, le poids IGP de l’arc est la distance IGP commune aux chemins encodés par le segment, et son délai est le temps de propagation maximum parmi ces chemins. Des segments d’adjacence peuvent s’ajouter, auquel cas les poids des arcs les représentant sont exactement égaux aux arcs initiaux qu’ils encodent. Seuls les arcs non-dominés sont conservés en cas d’arcs multiples.

Sur le graphe SR multi-métrique, les chemins originaux encodables en  $x$  segments sont représentés par des chemins de  $x$  arcs. En explorant le graphe SR de manière similaire à l’algorithme de Bellman-Ford, l’exploration des chemins itère donc sur le nombre de segments. Cette exploration permet de ne visiter que les chemins compatibles avec SR (en s’arrêtant à l’itération MSD) et de découvrir rapidement les chemins DCLC : les segments sont construits sur l’IGP, que DCLC cherche à minimiser. Si les coût IGP sont majoritairement alignés sur le délai, il est vraisemblable que la solution requière peu de segments.

## L'Algorithme Best2Cop : Déployer des Chemins Multi-contraints avec Segment Routing

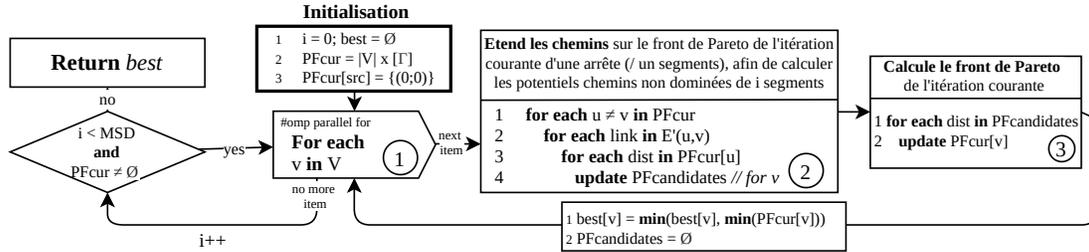


FIGURE 1: Diagramme représentant notre algorithme BEST2COP dans sa résolution de DCLC-SR

### 3 BEST2COP : Résoudre DCLC-SR sur le Graphe SR

Grâce au graphe SR, BEST2COP explore les chemins par nombre croissant de segments, i.e., par nombre croissant d'arcs, et mémorise toutes les distances non-dominées : leur nombre est borné par  $|V| \times \Gamma$  à chaque itération. BEST2COP est décrit à haut niveau sur la Fig. 1. Le front de Pareto de l'itération courante est mémorisé dans un tableau de taille  $|V| \times \Gamma$ , et contient initialement la distance  $(0,0)$  vers la source. A chaque itération  $i$ , BEST2COP parcourt tous les nœuds  $v$  depuis chaque nœud  $u$ . En combinant les distances vers  $u$  non-dominées découvertes à l'itération  $i - 1$  avec les poids des arcs  $(u, v)$ , BEST2COP calcule tous les nouveaux chemins de  $i$  segments vers  $v$ , *candidats* au front de Pareto. Bien que filtrés, certains de ces chemins candidats peuvent être dominés en fin d'itération. La vérification de leur appartenance au front de Pareto n'est faite qu'une fois que toutes les distances ont été calculées (boîte 3) pour amortir la complexité de mise à jour du front de Pareto ; cette vérification n'étant réalisée qu'une seule fois par itération et pas à chaque ajout/modification. Ces nouvelles distances sont à leur tour étendues d'un arc et ainsi de suite. Ce processus est répété MSD fois, afin d'explorer l'intégralité du front de Pareto 3D, mais uniquement celui des distances encodables avec SR : la solution à DCLC-SR est alors fournie vers l'ensemble des destinations.

Outre le maintien efficace du front de Pareto, les bonnes performances de BEST2COP proviennent également des structures utilisées. La taille du front étant bornée et prédictible grâce à la discrétisation, BEST2COP peut bénéficier de structure efficaces, par exemple des tableaux statiques de taille  $\Gamma$ . Cerise sur le gâteau, les itérations de la boîte 3 sont indépendantes (chaque itération explore des distances vers des nœuds  $v$  distincts), permettant de facilement paralléliser cette boucle. Le graphe SR Multi-Métrique étant a minima une clique, les charges des différents threads sont aisément distribuables de manière équitable.

BEST2COP possède une complexité de  $O(MSD \times |V|^2 \times L \times \Gamma)$ . Pour les  $|V|$  voisins de  $|V|$  nœuds, BEST2COP étend au pire  $\Gamma$  distances par  $L$  liens parallèles, le tout MSD fois. Ce coût peut être divisé par le nombre de threads utilisés. Dans l'évaluation, nous considérons  $\Gamma = 1000$  et  $L = 2$  sur le graphe SR (davantage que le nombre que nous avons observé en pratique).

### 4 Evaluation : BEST2COP est efficace en toutes circonstances

Dans cette dernière partie, nous évaluons les performances de BEST2COP selon trois options : une étude au pire, une étude sur graphes et valuation aléatoires, et enfin sur graphes et valuation réels. Le graphe SR étant inhérent à de nombreuses applications SR, nous ne prenons pas en compte son temps de création, le surplus ajouté par nos calculs pour la considération de métriques multiples étant marginal. Ces expériences sont faites sur un Intel(R) Core(TM) i7-8700B @ 3.20GHz avec 12 coeurs logiques.

Le scénario le plus défavorable consiste à devoir explorer l'ensemble de l'espace d'itération. BEST2COP se comporte alors comme s'il avait effectivement  $|V|^2 \times L \times c_1$  distances à étendre à chaque itération (avec  $L = 2$ ). La Fig. 2a présente une évaluation du temps d'exécution en fonction de  $|V|$  et du nombre de threads  $T$ . Pour  $|V| = 1000$ , le temps d'exécution atteint 75 secondes. Bien que raisonnable vu la nature extrême de l'expérience, cette valeur peut être réduite avec davantage de threads : le temps d'exécution est inférieur à 10 secondes avec  $T = 12$ . Le caractère parallélisable de BEST2COP a également été expérimenté sur une grille haute-performances : avec 30 coeurs, BEST2COP atteint une accélération de 23.

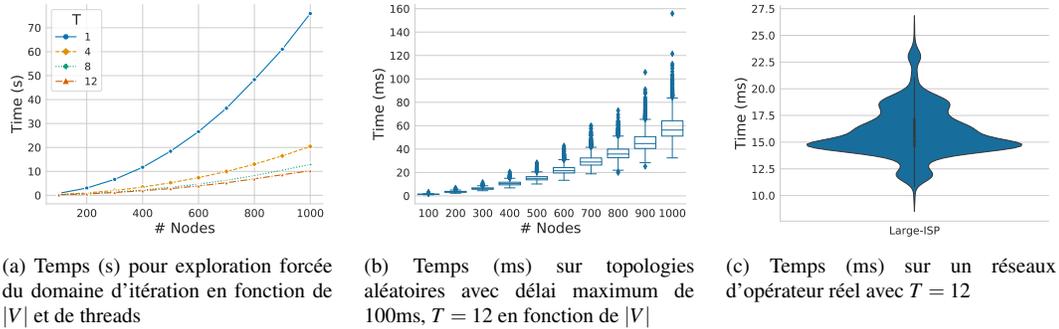


FIGURE 2: Evaluation du temps d'exécution sur différents cas avec  $\Gamma = 1000$

Dans la seconde expérience, sur base de graphes initiaux aléatoires en valuation et structure, nous produisons leur transformation en graphes SR. Nous utilisons le modèle d'Erdos-Rényi avec une densité de  $|V|/10$ . Les poids des liens sont inférés selon une loi de Zipf (avec une courbure de  $\log(9)/\log(10)$ ) :  $M_1$  est tirée entre 1 et  $10^3$ , et  $M_2$  entre 1 et  $10^8$ . Pour chaque valeur de  $|V|$  allant de 100 à 1000 avec un pas de 100, nous générons 30 graphes et lançons BEST2COP depuis 10% des nœuds. La figure 2b, nous montre le temps d'exécution en fonction de  $|V|$  (avec  $T = 12$ ). En excluant les temps les plus extrêmes, la plupart des temps d'exécution sont de l'ordre de 60ms. Bien qu'éloignées de la réalité, les caractéristiques structurelles des réseaux initiaux permettent de réduire les temps d'exécution même avec une valuation aléatoire.

Nous étudions pour finir les résultats sur un réseau réel de 1167 nœuds et 4408 liens avec valuation réelle. La figure 2c montre les résultats des exécutions de BEST2COP sur chacun des nœuds. Nous constatons que la majorité des temps se situe près de 15ms. Cette évaluation nous montre qu'en bénéficiant des propriétés structurelles et logiques (valuations) réelles des réseaux, BEST2COP dépasse rarement un temps d'exécution de 20ms. Notre algorithme est donc efficace pour des cas simples ou complexes grâce aux caractéristiques des valuations qui réduisent fortement le nombre de chemins non-dominés.

## 5 Conclusion & Perspectives

Segment Routing a ravivé l'intérêt des opérateurs pour l'ingénierie de trafic, en particulier pour le calcul de chemins DCLC. Nous proposons une solution massivement parallélisable résolvant efficacement DCLC pour un déploiement réel dans des domaines SR. En tirant parti des avantages de SR, de la fidélité des mesures du délai, ainsi qu'une exploration amortie du front de Pareto, BEST2COP résout DCLC-SR en moins de 20ms avec peu de threads (12) sur de larges graphes réalistes de plus de mille nœuds.

Des structures plus évoluées peuvent être envisagées pour permettre à BEST2COP de mieux appréhender une fidélité de mesure plus élevée. BEST2COP visitant l'intégralité du front de Pareto, son mode de fonctionnement peut également être étendu pour enregistrer toutes les distances explorées afin de résoudre d'autres problèmes connexes à DCLC en une seule exécution. Enfin, pour des réseaux partitionnés en aires, il semble intéressant de décomposer DCLC en plusieurs sous-instances afin de réduire encore davantage la complexité de BEST2COP sur de très gros réseaux.

## Références

- [Deb05] Kalyanmoy Deb. *Multi-Objective Optimization*, pages 273–316. Springer US, 2005.
- [FNP<sup>+</sup>15] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois. The segment routing architecture. In *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015.
- [GVRK18] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer. Unicast qos routing algorithms for sdn : A comprehensive survey and performance evaluation. *IEEE Communications Surveys Tutorials*, 20(1) :388–415, 2018.
- [LAM<sup>+</sup>20] J. R. Luttringer, T. Alfroy, P. Mérindol, Q. Bramas, F. Clad, and C. Pelsser. Computing delay-constrained least-cost paths for segment routing is easier than you think. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, 2020.