



Pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service

Ahmed Frikha, Gilles Bertrand, Samer Lahoud

► To cite this version:

Ahmed Frikha, Gilles Bertrand, Samer Lahoud. Pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service. [Rapport de recherche] PI 1935, 2009, pp.20. inria-00431635

HAL Id: inria-00431635

<https://inria.hal.science/inria-00431635>

Submitted on 12 Nov 2009

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.

Pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service

Ahmed Frikha^{*}, Gilles Bertrand^{**}, Samer Lahoud^{***}
ahmed.frikha@irisa.fr, gilles.bertrand@telecom-bretagne.eu, samer.lahoud@irisa.fr

Résumé: Garantir les performances d'un service à travers le réseau de plusieurs opérateurs est un problème complexe. Ce problème nécessite le calcul de chemins qui traversent plusieurs domaines et qui répondent aux différentes contraintes de qualité de service. De plus, les méthodes de calcul utilisées doivent respecter les contraintes de confidentialité et d'autonomie imposés par les domaines de différents opérateurs. Afin d'améliorer le temps de réponse, les opérateurs de réseaux mettent en place le pré-calcul de chemins. Le pré-calcul consiste à préparer à l'avance des chemins ou des segments de chemins qui sont utilisés ultérieurement dans le calcul de chemins de bout en bout. Dans ce travail, nous étudions le problème du pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service. Nous proposons une architecture qui met en place les solutions de précalcul inter-domaine et introduisons deux nouveaux algorithmes de pré-calcul ID-PPPA et ID-MEFPA. Nous évaluons les performances de ces deux algorithmes en les comparant avec des algorithmes exacts. Les résultats de simulation montrent que nos algorithmes permettent de satisfaire les demandes de calcul de chemins tout en respectant les contraintes de confidentialité entre les domaines.

Mots clés : Qualité de service, pré-calcul, contraintes multiples, routage inter-domaine.

Pre-computation of Inter-domain Paths Subject to Multiple QoS Constraints

Abstract: Ensuring the performances of a service through a several operators network is a complex problem. This problem requires the computation of paths that cross multiple domains and that meet the different quality of service constraints. In addition, the methods of computation that are used must meet the constraints of confidentiality and autonomy imposed by the domains of different operators. To improve response time, network operators use the path pre-computation method. Pre-computation consists of preparing in advance paths or segments of paths that are used later in the computation of paths crossing multiple domains. In this work, we investigate the problem of pre-computing inter-domain paths subject to multiple constraints of quality of service. We propose an architecture that implements inter-domain pre-computation solutions and introduce two new pre-computation algorithms ID-PPPA and ID-MEFPA. We evaluate the performance of these two algorithms by comparing them with exact algorithms. The simulation results show that our algorithms satisfy the computation requests while respecting the confidentiality constraints between domains.

Key-words: Quality of service, pre-computation, multiple constraints, inter-domain routing.

^{*} IRISA - Université Rennes 1

^{**} Institut Telecom ; Telecom Bretagne ; RSM.

^{***} IRISA - Université Rennes 1

1 Introduction

Un mécanisme primordial pour fournir de la qualité de service aux applications est le routage avec qualité de service. Le routage avec qualité de service consiste à calculer un chemin qui satisfait plusieurs contraintes de qualité de service. Il existe deux variantes pour le routage avec qualité de service. La première est le calcul à la demande qui consiste à lancer la procédure de recherche d'un chemin satisfaisant les contraintes dès la réception de la requête de qualité de service. La deuxième variante est le pré-calcul qui consiste à calculer à l'avance plusieurs chemins de qualité de service pour des classes de service prédéterminées.

Nous étudions dans ce document le problème de pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service. Nous identifions les limitations des solutions actuelles pour pré-calcul avec qualité de service. Nous introduisons un nouvel algorithme pour le pré-calcul intra-domaine avec qualité de service. Nous l'appelons PPPA (Primary Path based Precomputation Algorithm). Ensuite, nous proposons une solution inter-domaine basée sur le pré-calcul qui permet de garantir la qualité de service aux applications et de réduire le temps d'établissement de la connexion. Cette solution introduit deux nouveaux algorithmes : ID-PPPA (Inter-Domain PPPA) basé sur l'algorithme PPPA et ID-MEFPA (Inter-Domain MEFPA) basé sur l'algorithme de pré-calcul MEFPA [3].

Le reste de ce document est organisé comme suit. La section 2 décrit le problème général de pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service. La section 3 présente l'architecture globale et les blocs fonctionnels de la solution distribuée pour le problème de pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service. Dans la section 4 nous détaillons ces blocs fonctionnels. La section 5 décrit les simulations effectuées ainsi que les résultats obtenus. Nous concluons ce document par une analyse de la solution proposée et proposons des extensions aux solutions.

2 Présentation du problème

Dans cette section, nous présentons les notions et les concepts liés au routage multi-contrainte, ensuite nous détaillons les problèmes liés au routage inter-domaine, enfin nous présentons le pré-calcul.

2.1 Routage multi-contrainte

Le routage multi-contrainte consiste à trouver un chemin entre deux nœuds d'un réseau qui prend en considération les contraintes de qualité de service. Pour pouvoir trouver ce chemin, il faut connaître les métriques de qualité de service sur les liens du réseau. Les métriques de qualité de service peuvent être classées en trois types : les métriques concaves telles que la bande passante, les métriques additives telles que le temps de propagation et les métriques multiplicatives telles que le taux de perte. Les métriques multiplicatives peuvent être transformées en des métriques additives en utilisant une fonction logarithmique. Les métriques concaves peuvent être résolues en élaguant la partie du graphe du réseau qui ne vérifie pas ces métriques et par suite la recherche du chemin s'effectue sur le graphe résiduel. Ainsi dans cette étude nous nous intéressons uniquement aux métriques additives.

2.1.1 Formulation du problème

Soit un graphe $G(N, E)$, où N est l'ensemble des nœuds et E est l'ensemble des liens, et soit un chemin $P \in P_{SD}$: l'ensemble des chemins de S vers D dans $G(N, E)$.

Nous notons $w_i(P)$ le coût du chemin P pour la métrique additive i . Ainsi $w_i(P)$ est donné par : $w_i(P) = \sum_{e_j \in P} (w_i(e_j))$ où e_j appartient à l'ensemble des liens qui constituent le chemin P . Soit $\vec{W}(P) = (w_1(P), w_2(P), \dots, w_k(P))$ le vecteur coût du chemin P où k est le nombre de contraintes. Soit $\vec{C} = (c_1, c_2, \dots, c_k)$ le vecteur contrainte que le routage avec qualité de service vise à satisfaire.

Définition 1 : L'espace des métriques de qualité de service (QoSMS) est défini par $W^k = W_1 * W_2 * \dots * W_K$, où W_i est un ensemble de réels positifs inclus dans R_+ et $w_i(P) \in W_i$ pour tout chemin de $G(N, E)$.

Définition 2 : Un chemin P est dit faisable si et seulement si $\forall i \in 1..k, w_i(P) \leq c_i$.

Définition 3 : Un chemin P est dit non dominé dans l'ensemble de chemins P_{SD} si et seulement si aucun chemin $P' \in P_{SD}$ ne vérifie : $\forall i \in 1..k w_i(P') \leq w_i(P)$ et $\exists j \in 1..k$ tel que $w_j(P') < w_j(P)$.

Définition 4 : Nous définissons la fonction d'énergie comme étant la fonction qui transforme le vecteur coût d'un chemin en un coût scalaire.

Définition 5 : Un chemin P est dit optimal pour une fonction d'énergie L dans l'ensemble de chemins P_{SD} si est seulement

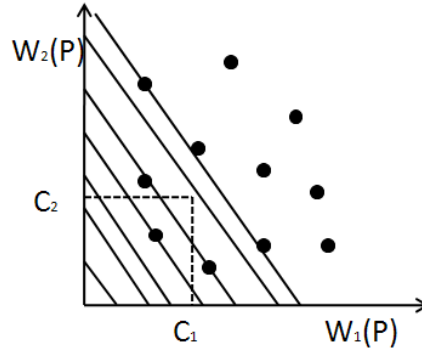


FIGURE 1 – Représentation des chemins de même coût dans le QoSMS à deux dimensions en utilisant une fonction linéaire d'énergie [6].

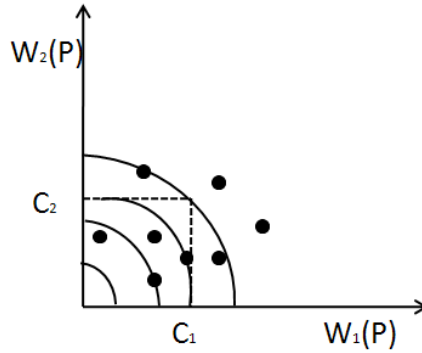


FIGURE 2 – Représentation des chemins de même coût dans le QoSMS à deux dimensions en utilisant la fonction non linéaire d'énergie L_q [6].

si aucun chemin $P' \in P_{SD}$ ne vérifie : $L(P') < L(P)$, avec $L(P)$ est le coût du chemin P pour la fonction L .

Définition 6 : Nous appelons un problème MCP (Multi-Constrained Path Problem) le problème qui consiste à trouver un chemin faisable entre un nœud source et un nœud la destination.

Définition 7 : Nous appelons un problème MCMOP (Multi Constrained and Multi objective Optimization Problem) le problème qui consiste à trouver le chemin optimal entre un nœud source et un nœud la destination qui minimise toutes les contraintes si un tel chemin existe.

Définition 8 : Nous appelons un problème MOOP (Multi-Objective Optimization Problem) le problème qui consiste à trouver l'ensemble de chemins non dominés entre un nœud source et un nœud la destination.

2.1.2 Les concepts du routage multi-contraainte

Le choix de la fonction d'énergie : Pour calculer le coût d'un chemin, il existe deux types de fonction : les fonctions linéaires et les fonctions non linéaires.

- Un exemple d'une fonction d'énergie linéaire est donné par la fonction linéaire de Jaffe [7] : $L(P) = \sum_{i=1}^k (d_i * w_i(p))$, où d_i est un coefficient. Ces fonctions d'énergie ne permettent pas toujours de résoudre le problème MCP. En effet, certains chemins ont un coût inférieur à celui du vecteur $\vec{C}$ mais ils ne sont pas forcément faisables. Ainsi, pour le routage exact avec qualité de service, nous n'avons pas intérêt à utiliser une fonction d'énergie linéaire. La figure 1 [6] représente la fonction de Jaffe dans l'espace de dimension 2 (le nombre de contraintes $k = 2$). Dans cette figure, chaque point de coordonnées $(w_1(P), w_2(P))$ représente un chemin P appartenant à P_{SD} et chaque droite représente l'ensemble des chemins de même coût. Nous remarquons que les points appartenant à la même droite ne satisfont pas toujours les deux contraintes à la fois.
- Un exemple d'une fonction d'énergie non linéaire est donné par : $L_q(P) = (\sum_{i=1}^k (\frac{w_i(P)}{c_i})^q)^{\frac{1}{q}}$ [6]. La figure 2 [6] représente l'ensemble des chemins de même coût calculé en utilisant cette fonction. Lorsque q tend vers l'infini cette fonction tend vers la fonction : $L_\infty(P) = \max_{i \in 1..k} (\frac{w_i(P)}{c_i})$ [6]. La fonction non linéaire L_∞ permet d'éliminer tous les chemins infaisables. En effet, tout chemin ayant un coût supérieur à 1 est infaisable. Par conséquent, un chemin retourné par un algorithme qui utilise cette fonction d'énergie est toujours faisable.

L'approche des K plus courts chemins : En utilisant des fonctions d'énergie non linéaires, un sous-chemin d'un chemin optimal n'est pas forcément un chemin optimal. Par conséquent, l'application de l'algorithme de Dijkstra avec une fonction d'énergie non linéaire ne permet pas toujours de résoudre le problème MCP même si ce problème admet une solution. L'approche des K plus courts chemins propose de mémoriser au niveau d'un nœud intermédiaire les K plus courts chemins et non pas uniquement le plus court chemin. Lorsque $K = \infty$, cet approche permet de résoudre le problème MCP lorsque le problème admet une solution. Afin réduire la complexité du calcul du chemin faisable, certaines heuristiques prennent K un entier positif. Cependant, ces heuristiques ne permettent pas toujours de résoudre le problème MCP même si un chemin faisable existe.

La non-dominance : Ce concept est très important pour le routage exact avec qualité de service. En effet, il permet de réduire l'espace de recherche des chemins en éliminant tous les chemins dominés de cet espace. Ainsi, ce concept ne considère que les chemins non dominés pour le calcul du chemin optimal. L'article [6] montre que même si nous excluons les chemins dominés le chemin optimal trouvé est le même chemin calculé. En effet, un chemin dominé ne peut pas être ni un chemin optimal ni un sous-chemin d'un chemin optimal. Par conséquent, l'utilisation de ce concept permet de trouver le chemin optimal tout en réduisant le temps de calcul puisque l'espace de recherche est réduit.

Le *Look-Ahead* : Ce mécanisme permet de réduire l'espace de recherche. En effet, au niveau d'un nœud intermédiaire n , un algorithme de routage qui met en œuvre ce concept utilise des informations sur les chemins qui mènent vers la destination à partir du nœud n . Ces informations permettent d'exclure des chemins non faisables de l'espace de recherche. Ce concept propose de calculer pour chaque contrainte l'arbre des plus courts chemins dont la racine est la destination. Pour cela, l'algorithme de Dijkstra est exécuté k fois où k est le nombre de contraintes. Le chemin obtenu pour chaque nœud n et pour chaque contrainte i est noté par $P_{n \rightarrow D; i}^*$, où D est la destination. Le vecteur $\vec{b}(n) = (b_1, \dots, b_k)$ est défini comme étant le vecteur dont les composantes sont données par $b_i = w_i(P_{n \rightarrow D; i}^*)$. Ce vecteur représente le meilleur coût pour atteindre la destination à partir du nœud n . Ainsi, un nouvel vecteur coût pour tout chemin $P_{S \rightarrow n}$ entre la source S et un nœud intermédiaire n peut être défini. Ce vecteur est donné par $\vec{W}(P_{S \rightarrow n}) + \vec{b}(n)$. Ainsi, si l'une des composantes de ce vecteur coût ne satisfait pas la contrainte, le chemin entre la source S et le nœud n est exclu du calcul. En effet, il ne peut pas être un sous-chemin d'un chemin faisable entre la source S et la destination D . Des fonctions additionnelles pour le *look-Ahead* peuvent être définies. Par exemple, au lieu d'appliquer Dijkstra pour chaque contrainte nous pouvons utiliser une fonction d'énergie linéaire telle que la fonction de Jaffe [7].

2.1.3 Solutions existantes pour le routage multi-contrainte

Des algorithmes qui permettent le calcul des solutions exactes, tel que l'algorithme SAMCRA [6], existent dans le cas intra-domaine. L'algorithme SAMCRA permet toujours de résoudre le problème MCP si ce problème admet une solution faisable. En plus, il permet toujours de trouver le plus court chemin puisqu'il mémorise au niveau de chaque nœud tous les chemins trouvés. Ainsi, SAMCRA permet de dépasser le problème de la non optimalité d'un sous-chemin. L'heuristique TAMCRA [6] de SAMCRA fixe le nombre maximal de chemins à mémoriser au niveau de chaque nœud. Ainsi, l'algorithme TAMCRA ne permet pas toujours de résoudre le problème MCP. Cependant, il est plus pratique pour les réseaux à grand nombre de nœuds étant donné sa complexité réduite par rapport à celle de l'algorithme SAMCRA. L'algorithme H-MCOP [18] est une heuristique permettant de trouver des chemins faisables pour le problème MCP. H-MCOP exécute l'algorithme de Dijkstra deux fois avec des modifications légères. Il utilise une fonction d'énergie non linéaire. Ces performances sont très proches de celles d'un algorithme exact.

2.2 Routage inter-domaine

Un domaine est un ensemble de nœuds interconnectés entre eux formant un système autonome (Autonomous System ou AS). Chaque domaine est interconnecté à d'autres domaines grâce à des nœuds appelés nœuds de bordure (Border Node ou BN), ces nœuds permettent la communication inter-domaine.

Chaque domaine est sous la responsabilité d'un opérateur. En pratique, la coopération entre les différents opérateurs est limitée. En effet, des informations sur la topologie et les ressources disponibles sur les liens ne peuvent pas être communiquées entre les différents opérateurs qui se trouvent en concurrence.

Un autre problème important lié au routage inter-domaine est le problème de passage à l'échelle. En effet, le calcul du chemin qui satisfait les contraintes traversant une séquence de domaine est beaucoup plus complexe que celui du calcul intra-domaine étant donné que le nombre de nœuds concernés par ce calcul est plus important.

2.3 Le pré-calcul de chemins

Le pré-calcul consiste à calculer plusieurs chemins qui satisfont différentes requêtes de qualité de service prédéterminées (classes de service) avant la réception de la requête de qualité de service. L'avantage de l'utilisation du pré-calcul par rapport au calcul à la demande est le gain en termes de temps de calcul. En effet, les chemins sont calculés à l'avance et par suite il suffit de trouver le meilleur chemin faisable parmi ceux-ci. Cependant, cette méthode nécessite d'effectuer des mises à jour

afin de prendre en considération les changements des poids sur les liens contrairement à la méthode de calcul à la demande où les poids utilisés pour le calcul du chemin faisable sont instantanés.

2.3.1 Algorithmes de pré-calcul intra-domaines multi-contraintes

Plusieurs algorithmes, tels que l'algorithme CDP [4] et l'algorithme MEFPA [3], utilisent le pré-calcul. L'algorithme MEFPA [3] utilise des fonctions d'énergie linéaires pour effectuer le pré-calcul, puis il mémorise les plus courts chemins correspondant aux différentes fonctions d'énergie utilisées. Cet algorithme sera présenté dans la section 3. L'algorithme CDP [4] est une extension du protocole RIP supportant le routage multi-contrainte. Pour réduire la taille de la table de routage, cet algorithme filtre les chemins dominés, ensuite il propose une technique d'agrégation des chemins pré-calculés basée sur la clusterisation. L'algorithme NM-MCP [5], utilise une approche hybride qui combine le pré-calcul au calcul à la demande. Cette technique consiste à calculer des chemins qui satisfont les contraintes sans nécessiter de phase de calcul à la demande ce qui permet de réduire considérablement le temps de réponse.

2.3.2 Algorithmes de pré-calcul inter-domaines multi-contraintes

L'algorithme pID-MCP [1,19] est un algorithme de pré-calcul multi-contrainte inter-domaine. Il enregistre tous les chemins non dominés entre les nœuds d'entrée du domaine et ceux du domaine voisin, ces chemins sont pré-calculés en appliquant l'algorithme SAMCRA.

L'heuristique kpID-MCP [1,19] de l'algorithme pID-MCP permet aussi le pré-calcul multi-contrainte inter-domaine. kpID-MCP se base sur TAMCRA au lieu de SAMCRA pour le pré-calcul des chemins.

3 Conception de la solution

Dans cette section nous proposons une architecture pour le pré-calcul inter-domaine multi-contrainte. Ensuite, nous présentons le fonctionnement interne des blocs fonctionnels qui constituent cette architecture ainsi que les algorithmes mis en œuvre dans chaque bloc.

3.1 Architecture pour le pré-calcul inter-domaine multi-contrainte

Dans cette sous-section, nous présentons l'architecture pour le pré-calcul inter-domaine multi-contrainte ainsi que les blocs fonctionnels qui la constitue. L'architecture comporte deux blocs : le bloc de pré-calcul intra-domaine et le bloc calcul inter-domaine (voir figure 3).

3.1.1 Bloc de pré-calcul intra-domaine

Ce bloc met en œuvre le pré-calcul intra-domaine. Au niveau de ce bloc, un algorithme de pré-calcul est exécuté permettant d'obtenir des chemins qui relient les nœuds de bordure du domaine à tous les autres nœuds du domaine. Ensuite, une procédure de filtrage des chemins dominés doit être exécutée.

3.1.2 Bloc de calcul inter-domaine

Le deuxième bloc permet la communication des résultats entre les domaines. Ce bloc comporte trois sous-blocs. Le premier sous-bloc permet la concaténation des résultats du pré-calcul et des résultats reçus du domaine précédant. Le deuxième sous-bloc permet le filtrage des chemins non faisables et des chemins dominés. Enfin, le troisième sous-bloc structure les résultats obtenus et les envoie au domaine suivant.

Le calcul inter-domaine est déclenché dès la réception d'une requête de qualité de service et s'arrête dès qu'un chemin faisable reliant la source et la destination est établi, ou bien lorsque aucun chemin faisable n'est trouvé. Dans ce dernier cas la requête est rejetée.

3.2 Fonctionnement interne des blocs fonctionnels

3.2.1 Pré-calcul intra-domaine

Algorithmes proposés pour le pré-calcul intra-domaine

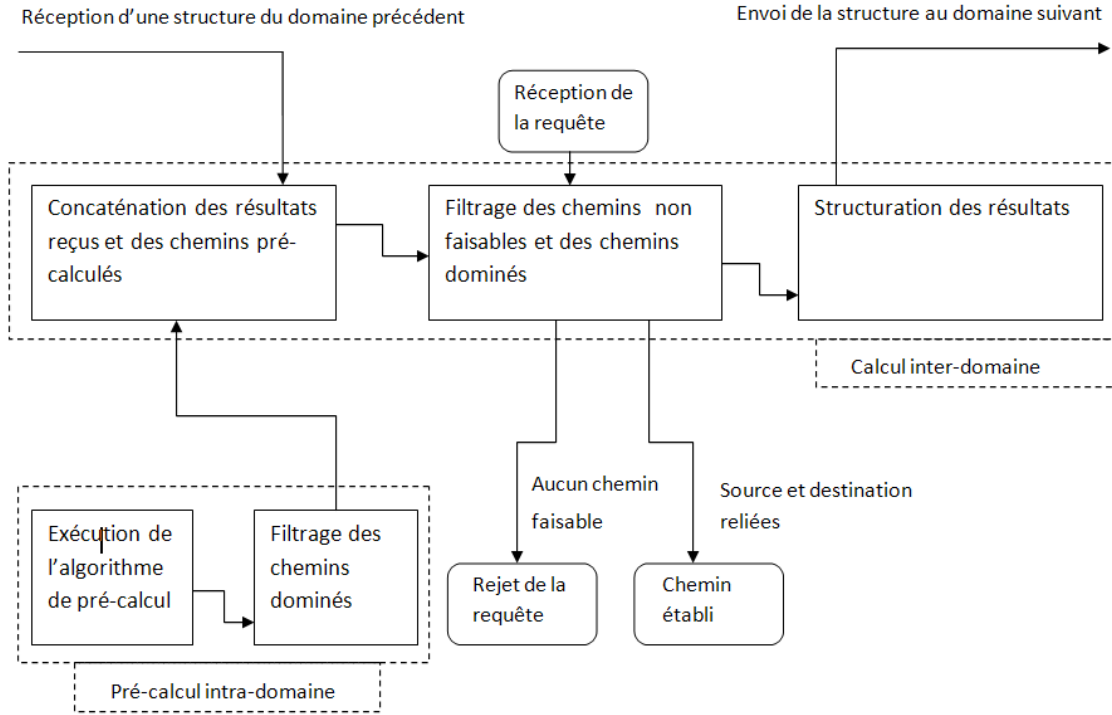


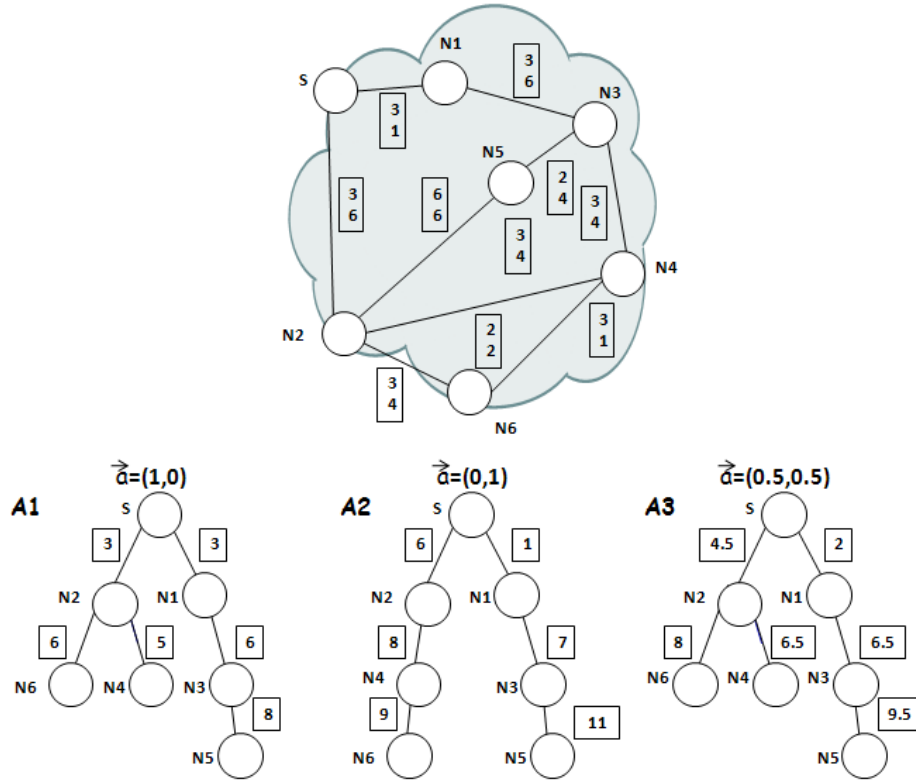
FIGURE 3 – Représentation des blocs fonctionnels de l'architecture

L'algorithme MEFFPA (Multi-constrained Energy Function based Precomputation Algorithm) [3], introduit dans la section 2, est un algorithme de pré-calcul avec qualité de service qui utilise plusieurs fonctions d'énergie linéaires pour calculer des chemins satisfaisant des contraintes de qualité de service prédéterminées.

MEFFPA calcule plusieurs arbres de plus courts chemins depuis chaque nœud du réseau. Ces arbres sont obtenus en exécutant l'algorithme de Dijkstra où le coût d'un lien est donné par la fonction d'énergie linéaire $g_a(e) = \sum_{i=1}^k a_i * w_i$ avec w_i est le coût du lien e pour la contrainte i et $\vec{a} = (a_1, a_2, \dots, a_k)$ est un vecteur coefficient de l'espace QoSMS. Soit P_a le chemin vérifiant $g_a(P_a) = \min \{g_a(P), P \in P_{SD}\}$. Ainsi P_a est le chemin qui minimise la fonction énergie g_a . L'ensemble des chemins P_a pré-calculés avec différents vecteurs coefficients $\vec{a}$ sera utilisé pour construire la table de routage avec qualité de service. Un algorithme pratique ne peut pas varier de façon continue la valeur de $\vec{a}$, pour cela MEFFPA utilise B vecteurs $\vec{a}$ uniformément distribués dans l'ensemble D^k où $D = \{\frac{0}{(b-1)}, \frac{1}{(b-1)}, \dots, \frac{b-1}{(b-1)}\}$ et b est un paramètre d'entrée de l'algorithme. À l'arrivée d'une requête $\vec{C}$, MEFFPA vérifie s'il existe un chemin P_a tel que $w_i(P_a) < c_i, \forall i \in 1..k$. Dans ce cas un chemin faisable est établi sinon la requête est rejetée. La complexité de MEFFPA dépend du paramètre b et du nombre de contraintes k . En effet, le coût d'exécution de MEFFPA [3] au niveau d'un seul nœud est donné par : $O(B * (n * \log(n) + m))$ qui est égal à B fois la complexité de Dijkstra où $B = C_{b+k-2}^{k-1}$, n étant le nombre de nœuds et m est le nombre de liens dans le domaine. La mise en œuvre de l'algorithme MEFFPA au niveau du bloc pré-calcul intra-domaine implique l'exécution de cet algorithme uniquement au niveau des nœuds de bordure du domaine. En effet, ce bloc propose de pré-calculer uniquement des chemins qui relient les nœuds de bordure aux autres nœuds du domaine. Par conséquent, le coût d'exécution de MEFFPA au niveau des nœuds de bordure d'un domaine est égale à : $O(N_{BN} * B * (n * \log(n) + m))$ où N_{BN} est le nombre de nœuds de bordure. La figure 4 illustre les arbres de plus courts chemins pré-calculés en utilisant l'algorithme MEFFPA. Dans cet exemple, le nombre de contraintes $k = 2$ et le paramètre $b = 3$. Par conséquent, l'ensemble $D = \{0, 0.5, 1\}$. Ainsi, nous obtenons trois arbres de plus courts chemins A_1 , A_2 et A_3 qui correspondent respectivement aux fonctions d'énergie de vecteurs coefficient $\vec{a} = (1, 0)$, $\vec{a} = (0, 1)$ et $\vec{a} = (0.5, 0.5)$.

Le nouvel algorithme de pré-calcul inter-domaine implémentant notre architecture et mettant en œuvre l'algorithme MEFFPA au niveau du bloc pré-calcul intra-domaine est appelé ID-MEFFPA (Inter-Domain MEFFPA).

L'algorithme PPPA (Primary Path based Precomputation Algorithm) Nous proposons dans ce document un nouvel algorithme simple de pré-calcul que nous appelons PPPA. Pour effectuer le pré-calcul, l'algorithme PPPA calcule les chemins primaires entre les nœuds du domaine. Nous définissons un chemin primaire P_i dans l'ensemble P_{SD} comme étant le plus court chemin entre un nœud source S et un nœud destination D qui minimise la contrainte i . L'ensemble des chemins primaires entre un nœud S et un nœud D est noté $P_{SDprimaires}$. Pour calculer cet ensemble l'algorithme PPPA exécute l'algorithme de Dijkstra k fois au niveau du nœud S , avec k est le nombre de contraintes, et en minimisant à chaque fois une contrainte. Autrement dit, à chaque exécution i , $i \in 1..k$, le coût d'un lien est donné par le poids de ce lien pour la contrainte i . À la réception d'une requête de qualité de service, cet algorithme vérifie si un chemin parmi les chemins primaires est faisable. Dans ce dernier cas, le chemin est retourné comme solution sinon la requête est rejetée.

FIGURE 4 – Exemple d'application de l'algorithme MEFPA avec $b = 3$ et $k = 2$

La complexité de PPPA dépend du nombre de contraintes k . En effet, le coût d'exécution de PPPA au niveau d'un seul nœud est donné par : $O(k * (n * \log(n) + m))$ qui est égal à k fois la complexité de l'algorithme de Dijkstra. La mise en œuvre de l'algorithme PPPA au niveau du bloc pré-calcul intra-domaine implique son exécution au niveau des nœuds de bordure du domaine. Par conséquent, le coût d'exécution de PPPA au niveau des nœuds de bordure d'un domaine est égal à : $O(N_{BN} * k * (n * \log(n) + m))$.

La figure 5 illustre l'ensemble des chemins primaires pré-calculés obtenus suite à l'exécution de l'algorithme PPPA. Dans cet exemple, le nombre de contraintes $k = 2$. Ainsi, nous obtenons deux arbres de plus courts chemins A_1 et A_2 . L'arbre A_1 minimise la première contrainte et l'arbre A_2 minimise la deuxième contrainte.

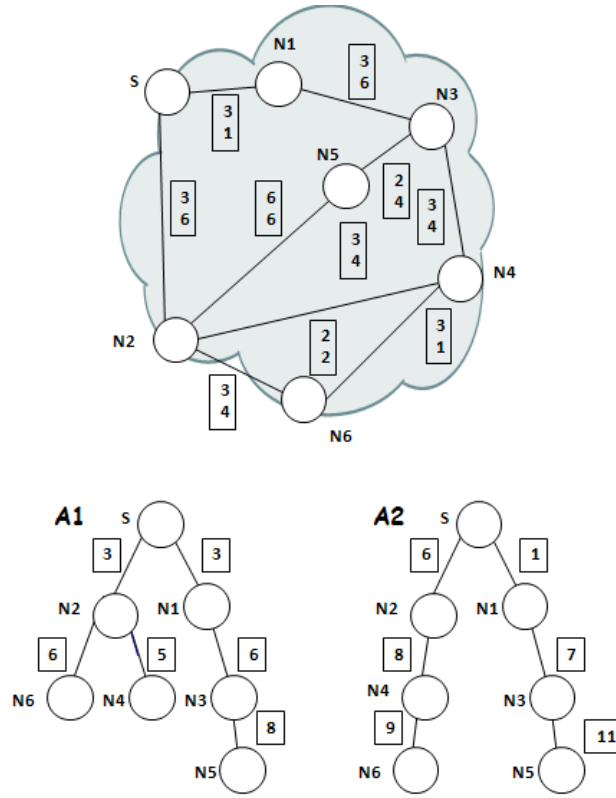
Le nouvel algorithme de pré-calcul inter-domaine implémentant notre architecture et mettant en œuvre l'algorithme PPPA au niveau du bloc pré-calcul intra-domaine est appelé ID-PPPA (Inter-Domain PPPA).

L'algorithme SAMCRA Comme introduit dans la section 2, SAMCRA est un algorithme exact de routage multi-contrainte. Il utilise la fonction de coût non linéaire $L_\infty(P) = \max_{i \in 1..k} \left\{ \frac{w_i(P)}{c_i} \right\}$. Ainsi pour trouver le chemin optimal entre un nœud source S et un nœud destination D , il suffit d'appliquer l'algorithme de calcul des K plus courts chemins. L'algorithme SAMCRA est un algorithme exact : il prend $K = \infty$ et mémorise tous les chemins calculés au niveau d'un nœud intermédiaire dont le coût est inférieur à 1. L'algorithme s'arrête quand le nœud à explorer est la destination ou lorsque tous les chemins sont exclus (coût supérieur à 1) et dans ce cas aucun chemin n'est faisable.

La complexité de SAMCRA dépend du nombre de contraintes de qualité de service k et du nombre de chemins mémorisés au niveau d'un nœud intermédiaire. La complexité au pire des cas est donnée par : $O(K_{max} * n * \log(K_{max} * n) + K_{max}^2 * k * m)$ [17] où K_{max} est le nombre maximum de chemins mémorisés au niveau d'un nœud. Une borne supérieure de K_{max} est donnée par : $K_{max} = \exp(n - 2)!$. Si de plus les contraintes sont de granularité finie, une deuxième borne supérieure pour K_{max} égale à $\prod_{i=1}^k \frac{c_i}{\max_j c_j}$ peut être définie. La mise en œuvre de l'algorithme SAMCRA au niveau du bloc pré-calcul intra-domaine implique son exécution au niveau des nœuds de bordure du domaine. Par conséquent, le coût d'exécution de SAMCRA au niveau des nœuds de bordure d'un domaine est égal à : $O(N_{BN} * (K_{max} * n * \log(K_{max} * n) + K_{max}^2 * k * m))$.

Basé sur notre architecture, l'algorithme de pré-calcul inter-domaine pID-MCP[1,19] met en œuvre l'algorithme SAMCRA pour effectuer le pré-calcul intra-domaine en supposant connu le vecteur contraintes de qualité de service $\vec{C}$ (classes de service prédéterminées).

L'algorithme TAMCRA TAMCRA est une heuristique de l'algorithme exact SAMCRA. Il utilise la même fonction d'énergie $L_\infty(P) = \max_{i \in 1..k} \left\{ \frac{w_i(P)}{c_i} \right\}$. Contrairement à SAMCRA, l'algorithme TAMCRA applique l'algorithme de calcul des K plus courts chemins mais en fixant le nombre de chemins à mémoriser à K . Ainsi, K sera un paramètre d'entrée

FIGURE 5 – Exemple d’application de l’algorithme PPPA avec $k = 2$

pour l’algorithme TAMCRA. Cette modification lui permet une complexité réduite par rapport à celle de SAMCRA. Cette complexité est donnée par : $O(K * n * \log(K * n) + K^2 * k * m)$. La mise en œuvre de l’algorithme TAMCRA au niveau du bloc pré-calcul intra-domaine implique son exécution au niveau des nœuds de bordure du domaine. Par conséquent, le coût d’exécution de TAMCRA au niveau des nœuds de bordure d’un domaine est égal à : $O(N_{BN} * (K * n * \log(K * n) + K^2 * k * m))$. Basé sur notre architecture, l’algorithme de pré-calcul inter-domaine kpID-MCP [1,19] met en œuvre l’algorithme TAMCRA pour effectuer le pré-calcul intra-domaine.

3.2.2 Calcul inter-domaine

Pour le calcul du chemin inter-domaine, nous avons proposé une version modifiée de la technique BRPC (Backward Recursive PCE-based Computation) [8]. Cette technique se base sur l’utilisation des PCE (Path Computation Element). Dans le paragraphe suivant nous présentons les PCE, puis nous décrivons la technique BRPC ainsi que son adaptation pour accomplir les fonctionnalités des sous-blocs du calcul inter-domaine (voir figure 3).

L’architecture PCE Comme le définit le RFC 4655 [9], le PCE est une entité capable de calculer un chemin à partir d’un graphe représentant le réseau. En plus, le calcul doit tenir compte des contraintes de qualité de service. Le calcul du chemin par un PCE se déclenche dès la réception de la requête d’un PCC (Path Computation Client). La communication entre ces deux entités est assurée par le protocole de communication PCC-to-PCE [10]. En absence d’information complète sur tous les domaines traversés, le PCE peut coopérer avec d’autres PCE pour le calcul du chemin inter-domaine.

La procédure BRPC Comme définie dans le RFC 5441 [8], la procédure BRPC est une technique de calcul de chemin inter-domaine basée sur la coopération des PCE. Dans cette procédure, le calcul du chemin inter-domaine commence à partir du domaine de la destination en supposant connue la séquence de domaines à traverser.

Chaque domaine possède un PCE responsable des calculs internes. Cependant le RFC 5441 ne spécifie pas l’algorithme multi-contrainte à implémenter au niveau du PCE. Par conséquent, chaque PCE utilise son propre algorithme pour le calcul interne.

La structure échangée entre les domaines est un arbre appelé VSPT (Virtual Shortest Path Tree). Cet arbre est construit en calculant les plus courts chemins en partant de la destination vers tous les nœuds de bordure permettant de joindre le domaine suivant dans la séquence de domaines à traverser.

La procédure est récursive, chaque domaine calcule cet arbre à partir du VSPT reçu du domaine précédent et des résultats du calcul interne et il envoie le nouveau VSPT au domaine suivant. Cet arbre ne contient que la destination (racine de l’arbre) et les nœuds de bordure du domaine actuel (feuilles) et les métriques de chaque chemin. Ainsi, cette technique permet d’assurer

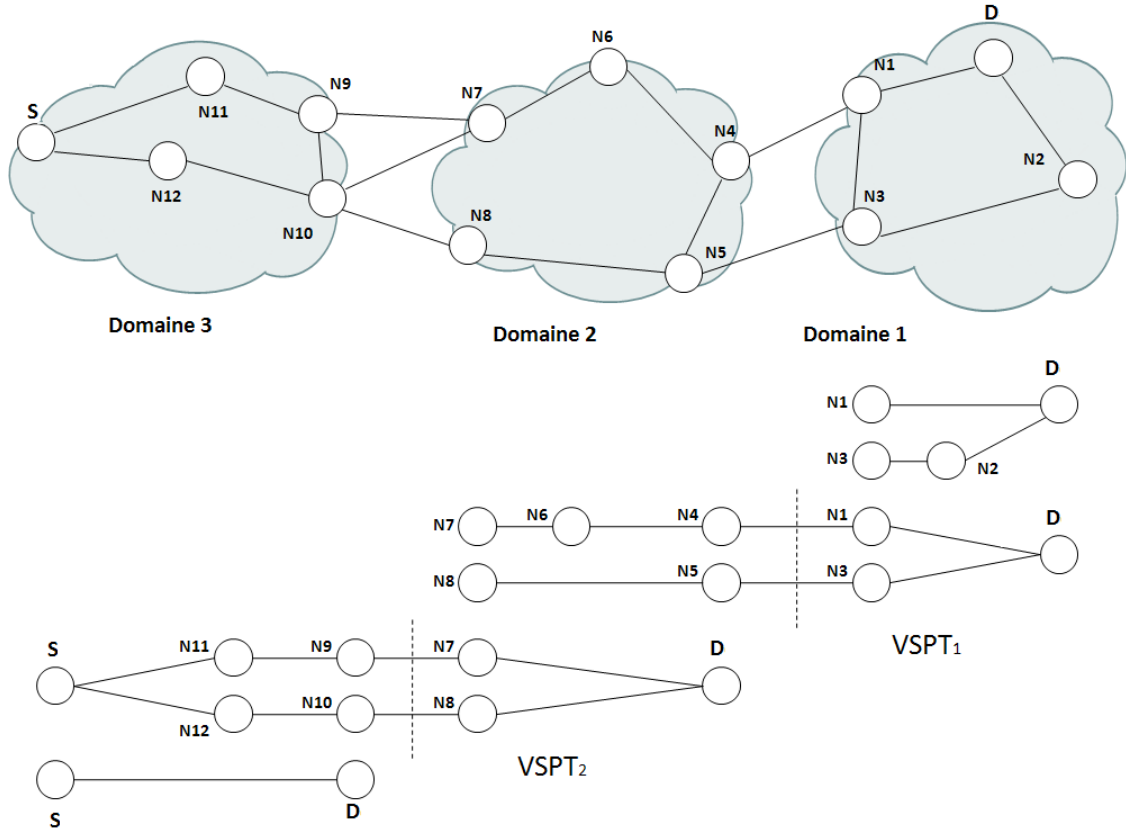


FIGURE 6 – Calcul inter-domaine avec la technique BRPC

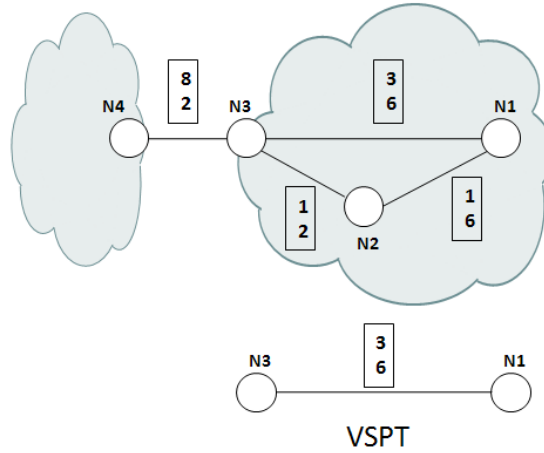


FIGURE 7 – Exemple de calcul inter-domaine où la structure VSPT ne permet pas de trouver de chemin faisable

la confidentialité entre les domaines. La figure 6 illustre un exemple de calcul inter-domaine en utilisant la procédure BRPC. Cependant, la technique BRPC ne permet pas toujours de trouver un chemin faisable inter-domaine même si un chemin existe. En effet, la structure échangée (VSPT) est un arbre, par conséquent, entre deux nœuds du VSPT il existe un seul chemin alors que entre deux nœuds il peut y avoir plusieurs chemins non dominés.

La figure 7 illustre le cas où la procédure BRPC ne permet pas de trouver de chemin faisable alors qu'un tel chemin existe. Dans cet exemple le vecteur contrainte $\vec{C}$ est égal à $(10, 10)$. Deux chemins permettent de relier le nœud N_1 au nœud N_3 . Le premier chemin est celui de coût égal à $(6, 3)$. Le deuxième chemin passe par le nœud N_2 et son coût est égal à $(2, 8)$. En supposant que l'algorithme SAMCRA est utilisé dans cet exemple pour effectuer le pré-calcul, seul le premier chemin est enregistré dans le VSPT pour être envoyé au domaine suivant. Comme le seul chemin faisable reliant le nœud N_1 et le nœud N_4 est celui qui passe par le nœud N_2 et le nœud N_3 , aucun chemin faisable n'est trouvé.

Les algorithmes proposés permettent le calcul de plusieurs chemins non dominés entre deux nœuds. Par conséquent, la structure VSPT n'est plus adaptée à nos algorithmes. Ainsi, nous proposons d'utiliser une *hiérarchie* [11] permettant de stocker entre deux nœuds tout les chemins pré-calculés faisables et non dominés. Ainsi, la structure communiquée entre un

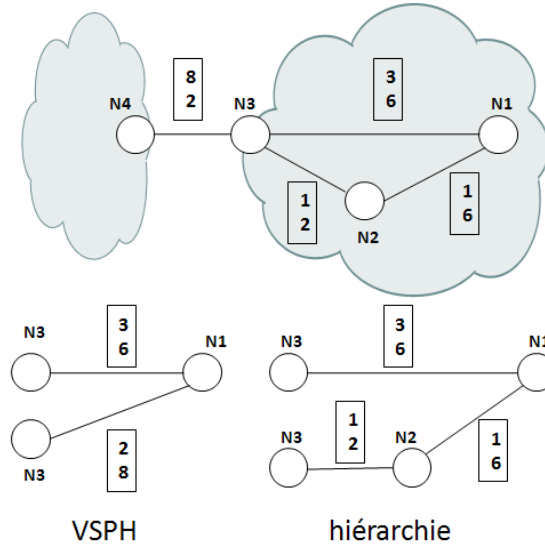


FIGURE 8 – La technique BRPC utilisant la structure VSPH

Domaine	Nombre de nœuds	Nombre de liens intra-domaines	Nombre de liens inter-domaines
Domaine 1	7	10	2
Domaine 2	12	17	2
Domaine 3	14	18	3
Domaine 4	17	24	3
Domaine 5	18	27	10

TABLE 1 – Caractéristiques de SYM-CORE

domaine et son successeur sera un VSPH (Virtual Shortest Path Hierarchy). La figure 8 montre que l'utilisation de cette structure au niveau de la procédure BRPC permet de trouver le chemin faisable non trouvé en utilisant la structure VSPT (voir figure 7).

4 Simulations et résultats

Dans cette section, nous évaluons les performances des deux algorithmes ID-PPPA et ID-MEFPA en les comparant aux algorithmes de pré-calcul inter-domaines pID-MCP et kpID-MCP. Nous commençons par la méthodologie de simulation adoptée, ensuite nous étudions les résultats obtenus.

4.1 Méthodologie de simulation

4.1.1 Topologies

Le choix de la topologie est très important lors de l'évaluation des algorithmes de routage. En effet, les performances de ces algorithmes dépendent des caractéristiques de la topologie telles que le nombre de nœuds et le degré de connectivité entre les nœuds. Pour cette raison, nous avons adopté deux types de topologies sur lesquelles nous avons effectué les simulations. Le premier type est représenté par des topologies en grille appelées Lattice. Ces topologies permettent d'étudier les performances de nos algorithmes dans un cas extrême de connectivité entre les nœuds. Nous distinguons deux types de topologies Lattice :

- LatticeFM(N, D) est une séquence de D domaines identiques. Elle représente le cas le plus extrême où chaque domaine est une matrice de N nœuds et de $E = 2\sqrt{N}(\sqrt{N} - 1)$ liens indirects. Chaque nœud du domaine est connecté à tous les nœuds du domaine suivant et à tous les nœuds du domaine précédent par un lien direct.
- LatticeSL(N, D) est similaire à la topologie LatticeFM(N, D). La seule différence entre ces deux topologies est la connectivité inter-domaine. En effet, dans LatticeSL deux domaines successifs sont connectés par un seul lien reliant le nœud en haut et à gauche du premier domaine au nœud en bas et à droite de son successeur.

Le deuxième type de topologie est représenté par la topologie réelle SYM-CORE [12]. Ce type de topologie permet d'étudier l'applicabilité de nos algorithmes dans un cas pratique. SYM-CORE est une topologie interzone composée de cinq domaines interconnectés. Le tableau 1 représente les caractéristiques de cette topologie.

Poids non corrélés	w_1 et $w_2 \in [10, 1023]$
Poids corrélés positivement	Si $w_1 \in [10, 511]$ alors $w_2 \in [10, 511]$ Si $w_1 \in [512, 1023]$ alors $w_2 \in [512, 1023]$
Poids corrélés négativement	Si $w_1 \in [10, 511]$ alors $w_2 \in [512, 1023]$ Si $w_1 \in [512, 1023]$ alors $w_2 \in [10, 511]$

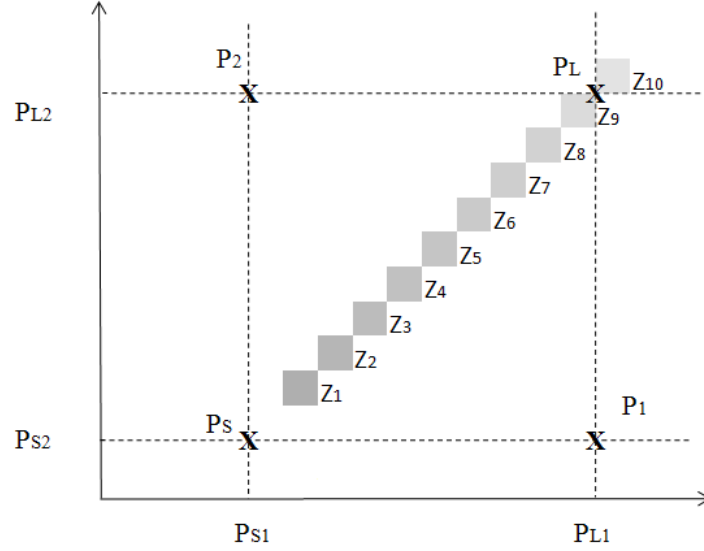
TABLE 2 – Génération des poids suivant le type corrélation avec un nombre de contraintes $k = 2$ 

FIGURE 9 – Les zones de tirage des contraintes de qualité de service

4.1.2 Génération des poids sur les liens

Les métriques de qualité de service sont des informations fournies par un protocole de routage. Ces métriques sont exprimées sur chaque lien par des poids afin de mesurer la qualité de chaque lien.

Dans nos simulations, nous considérons des métriques de qualité de service additives sans perdre de généralité comme expliqué dans la section 2. En plus, nous considérons que les poids sur les liens sont de granularité finie. En effet, le protocole de routage réserve un nombre d'octets fini pour le transfert de ces informations. De plus, les poids sur les liens peuvent être modélisés par une loi uniforme dans le réseau Internet [14]. Ainsi, les poids générés sur les liens dans toutes nos simulations suivent une loi uniforme $U[10,1023]$.

La corrélation des poids sur les liens influe sur les performances des algorithmes de routage multi-contrainte. Ainsi, afin d'étudier l'effet de la corrélation sur nos algorithmes, nous considérons tous les types de corrélation possibles lorsque le nombre de contraintes $k = 2$. Nous effectuons un tirage aléatoire du premier poids w_1 dans $[10,1023]$, puis nous effectuons un tirage aléatoire du deuxième poids w_2 en fonctions de w_1 . Plus précisément, nous divisons l'intervalle de tirage des poids $[10,1023]$ en deux sous-intervalles $I_1=[10,511]$ et $I_2=[512,1023]$. Ainsi, pour générer des poids corrélés positivement, si $w_1 \in I_1$ alors w_2 est tiré aléatoire dans I_1 , sinon w_2 est tiré aléatoirement dans I_2 . Alors que pour générer des poids corrélés négativement, si $w_1 \in I_1$ alors w_2 est tiré aléatoirement dans I_2 , sinon w_2 est tiré aléatoirement dans I_1 (voir tableau 2).

4.1.3 Génération des contraintes de qualité de service

Les performances d'un algorithme de routage avec qualité de service varient suivant la nature des contraintes de qualité de service. En effet, plus les contraintes sont strictes plus les taux de succès diminuent.

Dans nos simulations, nous étudions cet effet en parcourant l'espace de génération des contraintes de la zone la plus stricte vers la zone la plus lâche. La zone la plus stricte représente la partie de l'espace de génération des contraintes où l'existence d'un chemin faisable est peu probable. Alors que la zone la plus lâche représente l'espace de génération des contraintes où plusieurs chemins faisables peuvent exister. Pour parcourir l'espace de génération des contraintes, nous calculons l'ensemble des chemins primaires définis dans la section 2, et ceci pour chaque topologie, pour chaque distribution de poids sur les liens et pour chaque couple source et destination. Ensuite, nous calculons le point le plus strict $P_s = (p_{s1}, p_{s2}, \dots, p_{sk})$ et le point le plus lâche $P_l = (p_{l1}, p_{l2}, \dots, p_{lk})$ tels que $p_{si} = \min\{w_i(P); P \in P_{SDprimaires}\}$ et $p_{li} = \max\{w_i(P); P \in P_{SDprimaires}\} \forall i \in 1..k$. Enfin, nous définissons les zones de tirage de contraintes Z_j , $j \in 1..10$ tel que $Z_j = \{\vec{C} = (c_1, c_2, \dots, c_k); \forall i \in 1..k, p_{si} + (j + 1) * \frac{(p_{li} - p_{si})}{10} < c_i < p_{si} + (j + 2) * \frac{(p_{li} - p_{si})}{10}\}$. Les contraintes deviennent plus lâches en parcourant l'espace de tirage des contraintes de la zone Z_1 vers la zone Z_{10} . La figure 9 illustre les dix zones de tirage des contraintes.

4.1.4 Sélection de la séquence de domaines à traverser

Dans le réseau Internet, la plupart des domaines sont reliés entre eux à travers plusieurs séquences de domaines possibles. Cependant, le choix de la séquence de domaines, permettant de relier deux domaines, dépend de plusieurs facteurs tels que le facteur économique, la confidentialité du trafic entre les domaines concurrents, la politique de chaque domaine et les contrats de coopération entre les opérateurs assurant ou non la qualité de service. Ce problème dépasse le travail effectué dans ce document. Ainsi, dans nos simulations nous avons simplifié ce problème en considérant toujours la séquence de domaines la plus courte qui relie le domaine de la source et le domaine de la destination. En adoptant ce choix, nous simplifions le problème sans perdre de généralité en laissant le choix de la séquence de domaines à traverser aux opérateurs qui connaissent la liste des domaines avec lesquels ils coopèrent.

4.1.5 Taille de la séquence de domaines à traverser

Dans nos simulations, nous avons considéré des séquences de domaines de taille égale à trois dans les topologies de type Lattice. Dans la topologie SYM-CORE, la taille moyenne d'une séquence de domaines est comprise entre deux et trois. En effet, le domaine 5 de cette topologie est connecté à tous les autres domaines. L'article [15] justifie ce choix, en effet la taille moyenne d'une séquence de domaines dans le réseau Internet est égale à trois.

4.1.6 Métriques de performance

Deux métriques de performance sont utilisées pour évaluer les algorithmes de routage avec qualité de service :

- Le taux de succès SR (Success Rate) défini par le pourcentage de requêtes satisfaites parmi toutes les requêtes générées. Cette métrique permet de déduire la probabilité de trouver un chemin faisable pour une requête donnée.
- Le taux de succès relatif RSR (Relative Success Rate) défini par le pourcentage de requêtes satisfaites parmi toutes les requêtes qui sont satisfaites par un algorithme exact. Cette métrique permet de déduire la probabilité de trouver un chemin faisable sachant qu'au moins un chemin faisable existe.

Soit $P_F(A)$ l'ensemble des chemins faisables retournés par l'algorithme A . L'ensemble de tous les chemins faisables est noté P_F . Cet ensemble est égal à $P_F(E)$ l'ensemble des chemins faisables retournés par un algorithme exact E .

Afin d'évaluer les performances des algorithmes simulés, nous calculons aussi le pourcentage de chemins faisables retournés par chaque algorithme A défini par $NP(A) = \frac{Card(P_F(A))}{Card(P_F)} 1$. Cette métrique est calculée lorsque chaque algorithme trouve au moins un chemin faisable. Autrement dit, $Card(P_F(A)) \geq 1, \forall A$ un algorithme simulé. De cette manière, le taux de succès n'a pas d'influence sur cette mesure.

Pour évaluer la qualité des chemins retournés par un algorithme A , nous définissons deux autres métriques :

- Le coût $C(A)$ défini par la fonction $L_C(P) = \max_{i \in 1..k} \left(\frac{w_i}{c_i} \right)$ et utilisé par l'algorithme SAMCRA. Lorsque l'algorithme retourne plusieurs chemins, nous choisissons le coût le plus faible.
- Le coût $MC(A)$ défini par la fonction $L_{MC}(P) = \frac{1}{k} \sum_{i=1}^k \frac{w_i}{c_i}$. De même, lorsque l'algorithme retourne plusieurs chemins, nous choisissons le coût le plus faible.

Ces deux coûts sont calculés uniquement lorsque chaque algorithme simulé trouve au moins un chemin faisable, comme expliqué pour la métrique $NP(A)$.

4.2 Évaluation

Dans cette sous-section, nous étudions les résultats de simulation obtenus pour les algorithmes ID-MEFPA, ID-PPPA, pID-MCP et kpID-MCP. Pour l'algorithme kpID-MCP, le paramètre K qui représente le nombre maximum de chemins à mémoriser est égal à 4. Pour ID-MEFPA, le paramètre de MEFPA b est égal à 7.

Dans ce qui suit, chaque figure comporte plusieurs courbes mesurant la variation d'une métrique de performance en fonction des zones de tirage des contraintes de qualité de service. Chaque courbe est associée à un algorithme. Chaque courbe est représentée par dix points. Chaque point i d'une courbe représente la valeur moyenne de la métrique calculée dans la zone de tirage des contraintes Z_i . À chaque valeur moyenne est associée un intervalle de confiance à 95% (seuil de risque à 5%). Nous commençons notre étude par une analyse générale des résultats. Ensuite, nous évaluons l'effet de la connectivité inter-domaine, la corrélation des poids et l'augmentation du nombre de contraintes sur les performances des algorithmes.

4.2.1 Analyse générale des résultats

Les résultats obtenus dans les simulations montrent que l'algorithme ID-PPPA réalise de bonnes performances en dépit de sa simplicité et de sa complexité très réduite par rapport aux autres algorithmes. En effet, ses performances sont globalement très proches de celles de l'algorithme exact pID-MCP. Le seul cas où ses performances sont réduites par rapport aux autres est mesuré dans les topologies de type LatticeSL. Cependant, ces topologies représentent un cas très extrême de connectivité inter-domaine.

D'autre part, l'algorithme ID-MEFPA réalise de très bonnes performances qui dépassent dans la plus part des cas les

1. La fonction $Card(\xi)$ retourne le nombre d'éléments différents de l'ensemble ξ

performances de kpID-MCP et ID-PPPA. Cependant, sa complexité est élevée par rapport à celle de l'algorithme ID-PPPA surtout lorsque le nombre de contraintes k augmente.

4.2.2 L'effet de la connectivité inter-domaine

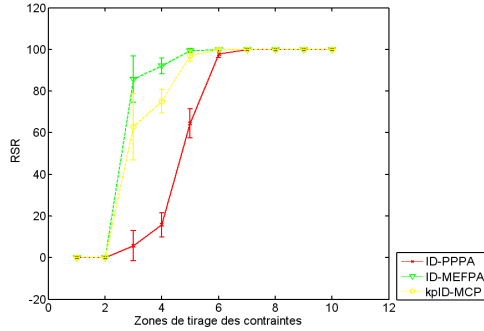


FIGURE 10 – Taux de succès relatif en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 3$ et des poids non corrélés

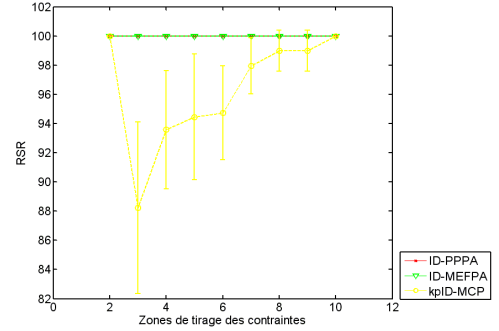


FIGURE 11 – Taux de succès relatif en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 3$ et des poids non corrélés

Les performances d'un algorithme de routage inter-domaine dépendent généralement du degré de connectivité inter-domaine. Pour évaluer cet effet sur nos algorithmes, nous considérons le taux de succès relatif mesuré dans les deux topologies LatticeFM(25,3) et LatticeSL(25,3). Dans la topologie LatticeFM(25,3) le nombre de liens qui relient deux domaines est égal à 625 liens, alors que dans la topologie LatticeSL(25,3) ce nombre est égal à 1.

Les figures 10 et 11 représentent le taux de succès relatif des algorithmes ID-PPPA, ID-MEFPA et kpID-MCP en fonction des contraintes de qualité de service avec un nombre de contraintes $k = 3$ et des poids non corrélés. Les simulations dans la figure 10 sont effectuées en utilisant la topologie LatticeSL(25,3) alors que dans la figure 11 les simulations sont effectuées en utilisant la topologie LatticeFM(25,3).

Nous remarquons que le taux de succès relatif des algorithmes ID-MEFPA et ID-PPPA sont très élevés pour la topologie LatticeFM(25,3) indépendamment de la zone de tirage des contraintes, alors que le taux de succès relatif de l'algorithme kpID-MCP est moins élevé. Dans la topologie LatticeSL(25,3), nous remarquons que ce taux diminue surtout pour l'algorithme ID-PPPA dans les zones de tirages des contraintes plus stricts. Ceci peut être expliqué par le fait que le nombre de chemins faisables retournés est beaucoup plus élevé dans la topologie LatticeFM(25,3). En effet, le nombre de chemins que contient le VSPH est très élevée étant donné le nombre élevé de nœuds de bordure. De plus, au niveau d'un domaine intermédiaire il existe plusieurs combinaisons possibles pour concaténer les chemins inclus dans le VSPH et les chemins pré-calculés. Ceci augmente la possibilité d'avoir plus de chemins faisables.

Les figures 12, 13 et 14 représentent le taux NP des chemins retournés en fonction des contraintes de qualité de service avec un nombre de contraintes $k = 2$ et des poids non corrélés. Les simulations dans les figures 12, 13 et 14 sont effectuées respectivement dans les topologies SYM-CORE, LatticeFM(25,3) et LatticeSL(25,3). Ces figures montrent que dans les topologies où la connectivité inter-domaine est élevée le taux NP est très élevé pour les algorithmes ID-MEFPA et ID-PPPA indépendamment des contraintes de qualité de service. Alors que pour kpID-MCP le taux NP est très faible et diminue lorsque les contraintes sont plus lâches. Cependant, pour les topologies où le degré de connectivité est moins élevé nous remarquons que ID-PPPA réalise la valeur la moins faible du taux NP . Ceci peut être expliqué par le fait que la borne supérieure du nombre de chemins calculés par ID-PPPA au niveau d'un domaine intermédiaire est égal à $k * NB$ avec NB le nombre de nœuds de bordure qui mènent vers le domaine suivant. Alors que cette borne est égale à $C_{b+k-2}^{k-1} * NB$ pour l'algorithme ID-MEFPA et égale à $K * NB$ pour l'algorithme kpID-MCP. D'autre part, les chemins calculés par l'algorithme kpID-MCP sont tous faisable puisque cet algorithme suppose connaître les contraintes de qualité de service. Ceci explique le fait que le taux NP de l'algorithme kpID-MCP est élevé par rapport à ceux des algorithmes ID-MEFPA et ID-PPPA dans les topologies dont la connectivité inter-domaine est très faible. Pour la topologie SYM-CORE, le taux NP diminue pour les algorithmes ID-PPPA, ID-MEFPA et kpID-MCP lorsque les contraintes sont de plus en plus lâches. En effet, l'augmentation du nombre de chemins faisables retournés par l'algorithme exact est plus rapide que celles de ces algorithmes. Cependant, pour les contraintes générées dans la zone de tirage la plus lâche, nous remarquons une augmentation du taux NP pour les algorithmes ID-MEFPA et ID-PPPA. En effet, dans cette zone tous les chemins pré-calculés par ces deux algorithmes sont faisables. Ainsi, le taux NP augmente pour les algorithmes ID-MEFPA et ID-PPPA dans cette zone. Ce taux continue à diminuer pour l'algorithme kpID-MCP dans cette zone. En effet, le nombre de chemins pré-calculés par cet algorithme est très limité.

4.2.3 L'effet de la corrélation des poids

La corrélation des poids influe sur les performances des algorithmes de routage multi-contrainte. Pour évaluer cet effet sur les performances de nos algorithmes, nous considérons le taux de succès mesuré dans la topologie SYM-CORE avec des

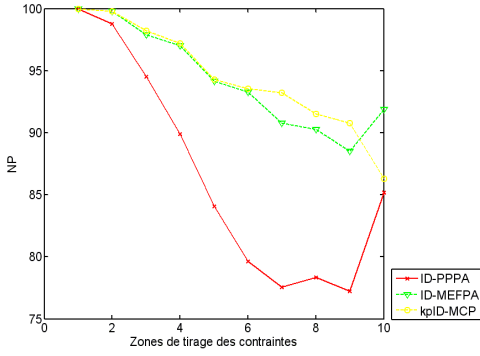


FIGURE 12 – NP en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids non corrélés

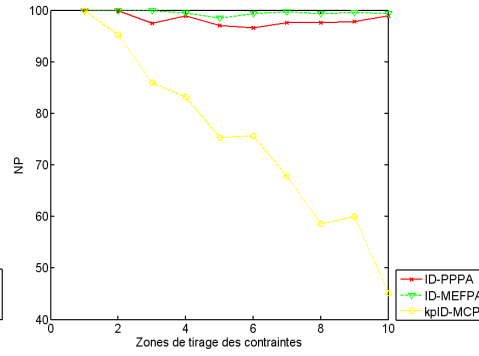


FIGURE 13 – NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 2$ et des poids non corrélés

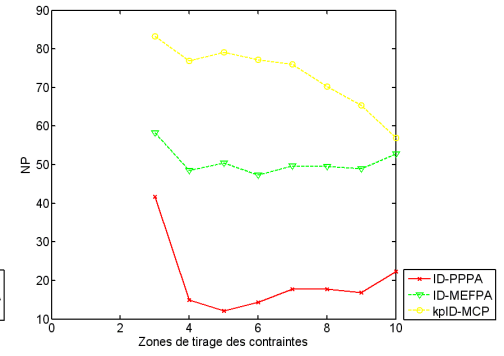


FIGURE 14 – NP en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids non corrélés

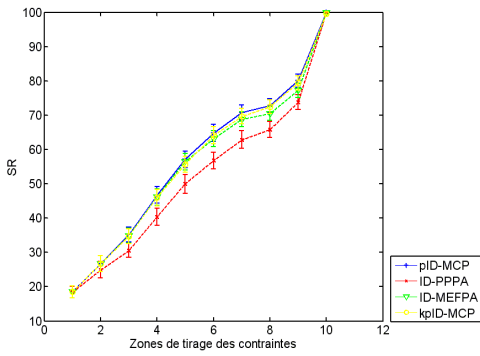


FIGURE 15 – SR en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids non corrélés

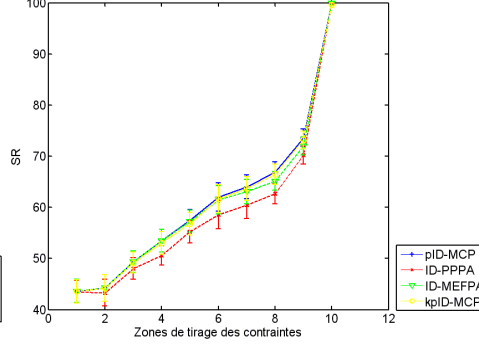


FIGURE 16 – SR en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids corrélés positivement

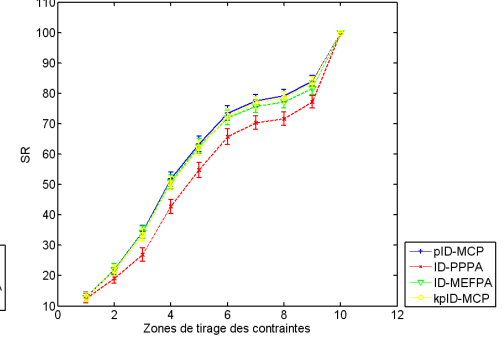


FIGURE 17 – SR en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids corrélés négativement

poids corrélés positivement, des poids corrélés négativement et des poids non corrélés.

Les figures 15, 16 et 17 représentent le taux de succès des algorithmes ID-MEFPA, ID-PPPA, pID-MCP et kpID-MCP avec un nombre de contraintes $k = 2$. Les simulations dans les figures 15, 16 et 17 sont respectivement effectuées avec des poids non corrélés, corrélés positivement et corrélés négativement.

Nous remarquons que dans les zones de tirage des contraintes les plus strictes, le taux de succès est plus élevé avec des poids corrélés positivement qu'avec des poids non corrélés. Il est encore plus faible pour des poids corrélés négativement. Ceci s'explique par le fait que les algorithmes de routage n'arrivent pas à satisfaire les deux contraintes à la fois. En effet, en cherchant à minimiser une contrainte, ils maximisent l'autre.

Cependant, nous remarquons que lorsque les contraintes sont plus lâches, le taux de succès avec des poids corrélés négativement est plus élevé qu'avec des poids corrélés positivement. En effet, avec des poids corrélés négativement, lorsque typiquement l'un des poids est tiré dans l'intervalle $[10, 511]$, l'autre poids est tiré dans le second intervalle. Soumis à des contraintes lâches, les algorithmes de routage n'ont plus qu'à satisfaire la contrainte dont le poids de la métrique de qualité de service correspondante est tiré dans l'intervalle $[512, 1023]$.

Nous remarquons aussi que le taux de succès des algorithmes ID-PPPA, ID-MEFPA, pID-MCP et kpID-MCP sont presque égaux dans chacun des cas de corrélations des poids.

Pour évaluer l'impact de la corrélation des poids sur la qualité des chemins retournés, nous considérons le coût MC mesuré dans la topologie SYM-CORE où le nombre de contraintes $k = 2$, avec des poids non corrélés, corrélés positivement et corrélés négativement (figures 18, 19 et 20).

Nous remarquons que la qualité des chemins est meilleure lorsque les poids sont corrélés négativement. En effet, le coût moyen MC diminue étant donné que le poids de l'une des métriques de qualité de service est tiré dans l'intervalle $[10, 512]$, ce qui n'est pas garanti dans les autres cas de corrélations des poids.

4.2.4 L'effet de l'augmentation du nombre de contraintes

Pour étudier l'effet de l'augmentation du nombre de contraintes sur les performances de nos algorithmes, nous considérons le taux de succès mesuré avec un nombre de contraintes $k = 2$ puis avec $k = 4$. Les figures 21 et 22 représentent ce taux en fonction des zones de tirage des contraintes dans la topologie LatticeSL(25,3) avec des poids non corrélés et un nombre de contraintes respectivement égal à 2 et 4.

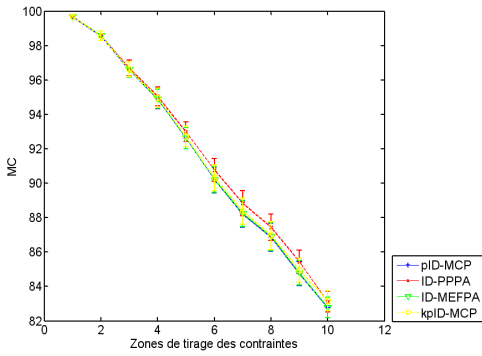


FIGURE 18 – MC en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids non corrélés

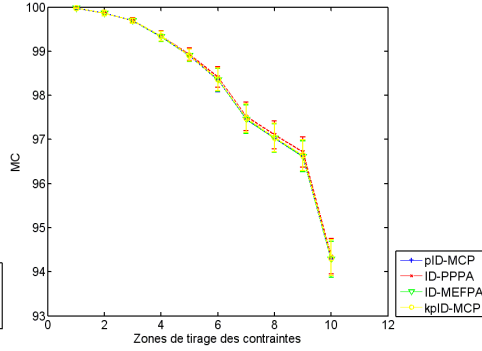


FIGURE 19 – MC en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids corrélés positivement

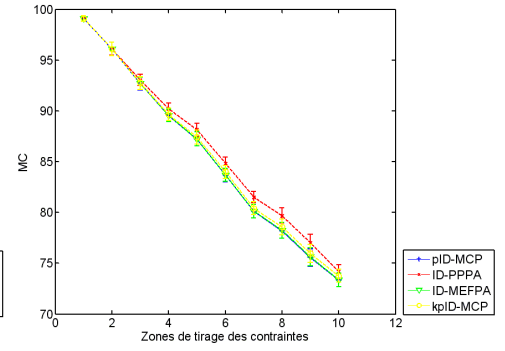


FIGURE 20 – MC en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 2$ et des poids corrélés négativement

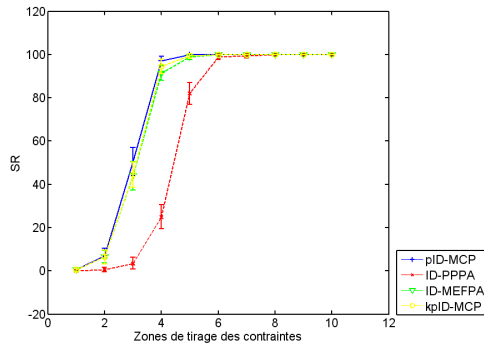


FIGURE 21 – SR en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids non corrélés

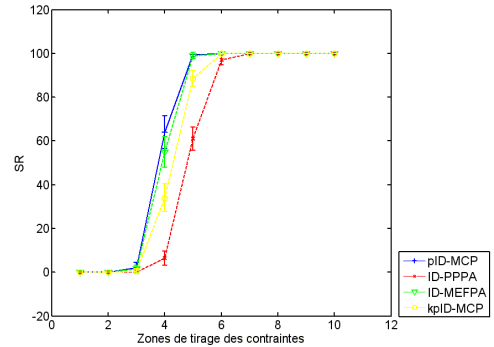


FIGURE 22 – SR en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

Nous remarquons que le taux de succès des algorithmes ID-MEFPA, ID-PPPA, pID-MCP et kpID-MCP diminue en général lorsque k augmente. Cependant, la variation de l'éloignement de ID-MEFPA et de ID-PPPA par rapport à l'algorithme exact pID-MCP n'est pas très importante en comparant à celle de kpID-MCP. Ceci peut être expliqué par l'augmentation du nombre de chemins pré-calculés par ID-MEFPA et ID-PPPA. En effet le nombre de chemins pré-calculés par ID-MEFPA et ID-PPPA dépend du nombre de contraintes : plus le nombre de contraintes augmente plus le nombre de chemins pré-calculés augmente. Ce qui permet d'adapter les performances de ces deux algorithmes en fonction du nombre de contraintes. Les figures 13, 23 et 24 illustrent l'invariance du taux NP pour les algorithmes ID-MEFPA et ID-PPPA lorsque le nombre de contraintes k augmente de 2 à 4 dans la topologie LatticeFM(25,3) avec des poids non corrélés, alors que le taux NP de kpID-MCP diminue de manière très importante pour $k = 3$ et $k = 4$.

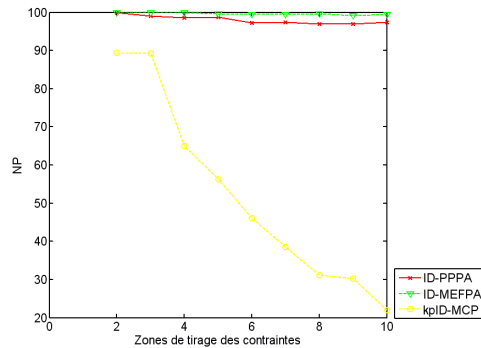


FIGURE 23 – NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 3$ et des poids non corrélés

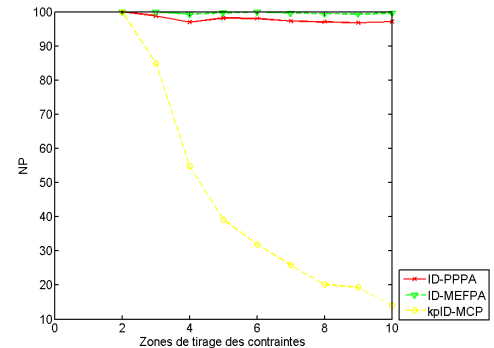


FIGURE 24 – NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

5 Conclusion

Ce document étudie le problème de pré-calcul de chemins inter-domaines soumis à plusieurs contraintes de qualité de service.

D'abord, nous avons étudié les algorithmes de routage intra-domaine avec qualité de service utilisant le pré-calcul et nous avons proposé un nouvel algorithme de pré-calcul intra-domaine nommé PPPA.

Ensuite, nous avons proposé une architecture permettant le routage inter-domaine avec qualité de service tout en utilisant le pré-calcul.

Enfin, nous avons proposé deux algorithmes de pré-calcul inter-domaines avec qualité de service. Le premier, appelé ID-PPPA, est une extension de l'algorithme PPPA pour un contexte inter-domaine. Le deuxième, appelé ID-MEFPA, étend l'algorithme de pré-calcul intra-domaine MEFPA.

Les simulations sont effectuées sur deux types de topologies. Le premier représente un cas extrême de connectivité, et le deuxième représente des topologies réelles. Nous remarquons que nos algorithmes réalisent de bonnes performances très proches de celles de l'algorithme exact de pré-calcul inter-domaine pID-MCP tout en réduisant la complexité des calculs. Ces performances sont meilleures que celles de l'algorithme kpID-MCP dans les topologies dont la connectivité inter-domaine est élevée. De plus, nos algorithmes adaptent leurs performances en fonction du nombre de contraintes de qualité de service.

Comme perspectives futures de ce travail, nous proposons d'étudier la stabilité de nos algorithmes face aux changements dynamiques des états des liens. Ce qui permet d'une part de déduire la validité des chemins pré-calculés après un changement possible des poids sur les liens, et d'autre part de proposer un seuil pour chaque algorithme à partir duquel il faut refaire le pré-calcul.

Table des matières

1	Introduction	2
2	Présentation du problème	2
2.1	Routage multi-contrainte	2
2.1.1	Formulation du problème	2
2.1.2	Les concepts du routage multi-contrainte	3
2.1.3	Solutions existantes pour le routage multi-contrainte	4
2.2	Routage inter-domaine	4
2.3	Le pré-calcul de chemins	4
2.3.1	Algorithmes de pré-calcul intra-domaines multi-contraintes	5
2.3.2	Algorithmes de pré-calcul inter-domaines multi-contraintes	5
3	Conception de la solution	5
3.1	Architecture pour le pré-calcul inter-domaine multi-contrainte	5
3.1.1	Bloc de pré-calcul intra-domaine	5
3.1.2	Bloc de calcul inter-domaine	5
3.2	Fonctionnement interne des blocs fonctionnels	5
3.2.1	Pré-calcul intra-domaine	5
3.2.2	Calcul inter-domaine	8
4	Simulations et résultats	10
4.1	Méthodologie de simulation	10
4.1.1	Topologies	10
4.1.2	Génération des poids sur les liens	11
4.1.3	Génération des contraintes de qualité de service	11
4.1.4	Sélection de la séquence de domaines à traverser	12
4.1.5	Taille de la séquence de domaines à traverser	12
4.1.6	Métriques de performance	12
4.2	Évaluation	12
4.2.1	Analyse générale des résultats	12
4.2.2	L'effet de la connectivité inter-domaine	13
4.2.3	L'effet de la corrélation des poids	13
4.2.4	L'effet de l'augmentation du nombre de contraintes	14
5	Conclusion	16

Références

- [1] G. Bertrand, S. Lahoud, M. Molnár, and G. Texier. Inter-Domain Path Computation with Multiple Constraints. In *IRISA*, Tech. Rep. 1902, 2008, <http://hal.inria.fr/inria-00319401>.
- [2] H. Elazhary, and S. Gokhale. Integrating path computation and precomputation for quality-of-service provisioning. In *Proceedings of the Ninth international Symposium on Computers and Communications*, 2004.
- [3] Y. Cui, K. Xu, and J. Wu. Precomputation for multiconstrained QoS routing in high-speed networks. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003.
- [4] Y. Cui, and J. Wu. Clustering-Based Distributed Precomputation for Quality-of-Service Routing. In *International Conference on Computational Science*, 2005.
- [5] Y. Zheng, T. Korkmaz, W. Dou, and J. Tian. Highly responsive and efficient QoS routing using pre- and on-demand computations along with a new normal measure. In *Computer Networks*, volume 50, issue 18, 2006.
- [6] P. Van Mieghem and F. A. Kuipers. Concepts of exact QoS routing algorithms. *IEEE/ACM Transactions on Networking*, 12(5) :851-864, 2004.
- [7] J. M. Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, 14 :95-116, 1984.
- [8] J.P. Vasseur, R. Zhang, N. Bitar, and J.L. Le Roux. RFC 5441 : A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths. IETF, April 2009.
- [9] A. Farrel, J. P. Vasseur, and G. Ash. RFC 4655 : A Path Computation Element (PCE)-Based Architecture. IETF, August 2006.

- [10] J. Ash, J.L. Le Roux. RFC 4657 : Path Computation Element (PCE) Communication Protocol. IETF, September 2006.
- [11] M. Molnar. Hierarchies for Constrained Partial Spanning Problems in. Graphs. In *IRISA*, Tech. Rep. 1900, 2008.
- [12] S. Dasgupta, JC. de Oliveira, JP. Vassuer. Performance Analysis of Inter-Domain Path Computation Methodologies draft-dasgupta-ccamp-path-comp-analysis-02. IETF, July 11, 2008.
- [13] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1) :2-16, Feb. 2004.
- [14] P. Van Mieghem, G. Hooghiemstra, and R. van der Hofstad. A scaling law for the hop count in internet. In *Delft University of Technology*, Technical report, 2000.
- [15] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 638-647, 2002.
- [16] H. De Neve and P. Van Mieghem. TAMCRA : a tunable accuracy multiple constraints routing algorithm. In *Computer Communications*, 23(7) :667-679, 2000
- [17] F. A. Kuipers and P. Van Mieghem. QoS routing : Average complexity and hopcount in m dimensions. In *Quality of Future Internet Services*, 2001
- [18] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 834-843, 2001.
- [19] G. Bertrand, S. Lahoud, M. Molnár, and G. Texier. Inter-Domain Path Computation with Multiple QoS Constraints, In *Recent Advances in Providing QoS and Reliability in the Future Internet Backbone*, Ning Wang, George Pavlou, and Mohamed Boucadair (Ed.), Nova Science Publisher, Expected publication in 2010.

Annexe A

Le taux de succès

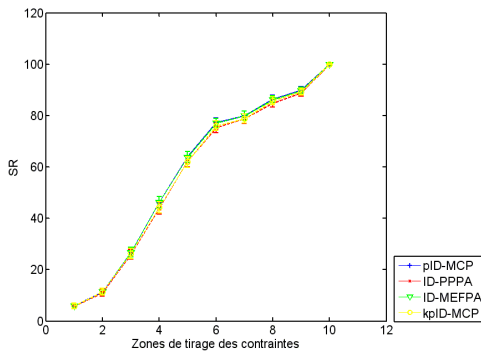


FIGURE 25 – SR en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 4$ et des poids non corrélés

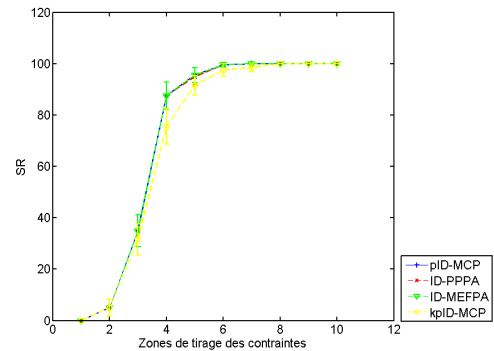


FIGURE 26 – SR en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

Les figures 25 et 26 illustrent la variation du taux de succès en fonction des contraintes de qualité de service respectivement dans les topologies SYM-CORE et LatticeFM(25,3), avec un nombre de contraintes $k = 4$. La comparaison de ces deux figures illustre l'effet de la connectivité inter-domaine sur le taux de succès. En effet, le taux de succès mesuré en utilisant la topologie LatticeFM(25,3) croît plus rapidement que celui mesuré en utilisant la topologie SYM-CORE lorsque les contraintes sont de plus en plus lâches.

Les figures 27 et 28 illustrent la variation du taux de succès en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) respectivement avec des poids corrélés positivement et des poids corrélés négativement, avec un nombre de contraintes $k = 2$. La comparaison des deux figures illustre l'effet de la corrélation des poids sur le taux de succès. En effet, pour les contraintes les plus strictes, le taux de succès mesuré avec des poids corrélés négativement est largement inférieur à celui mesuré avec des poids corrélés positivement. Alors que, pour les contraintes les plus lâches, le taux de succès mesuré avec des poids corrélés négativement est largement supérieur à celui mesuré avec des poids corrélés positivement.

Le taux NP

Les figures 29 et 30 illustrent le taux NP en fonction des contraintes de qualité de service dans la topologie SYM-CORE respectivement avec un nombre de contraintes $k = 3$ et $k = 4$. La comparaison des deux figures illustre l'invariance du

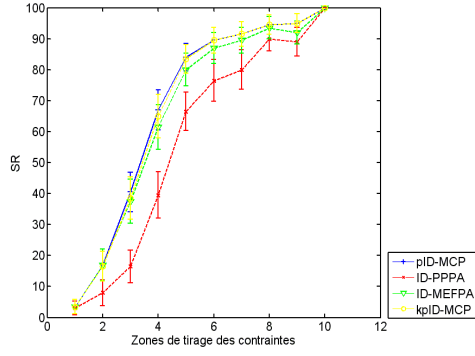


FIGURE 27 – SR en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés positivement

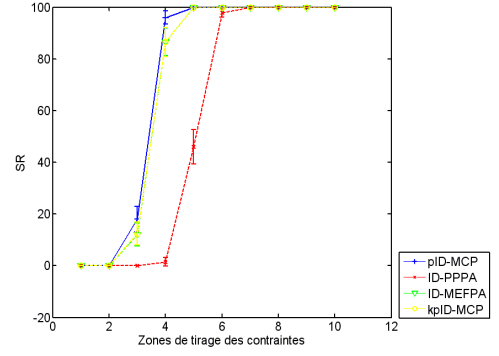


FIGURE 28 – SR en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés négativement

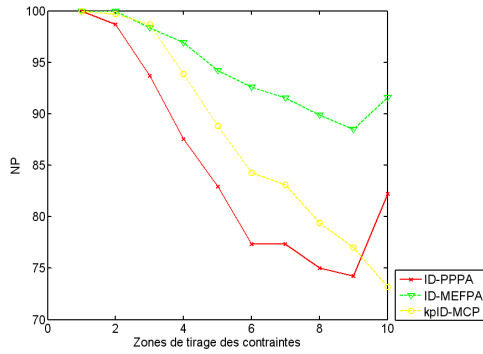


FIGURE 29 – NP en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 3$ et des poids non corrélés

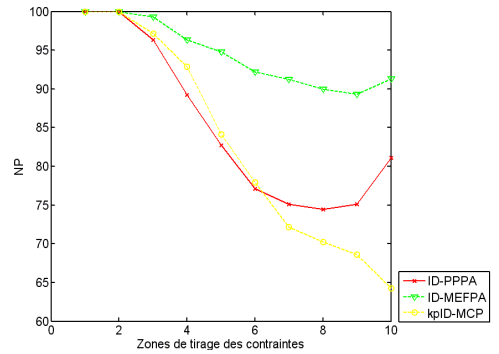


FIGURE 30 – NP en fonction des contraintes de qualité de service dans la topologie SYM-CORE avec un nombre de contraintes $k = 4$ et des poids non corrélés

taux NP pour les algorithmes ID-MEFPA et ID-PPPA, alors que pour l'algorithme kpID-MCP ce taux diminue lorsque k augmente.

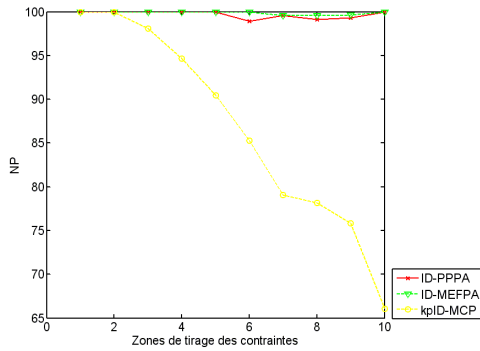


FIGURE 31 – NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés positivement

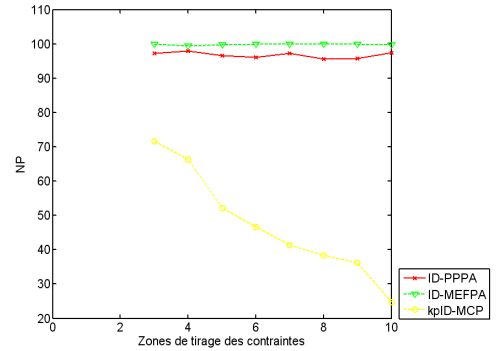


FIGURE 32 – NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés négativement

Les figures 31 et 32 illustrent la variation du taux NP en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) respectivement avec des poids corrélés positivement et des poids corrélés négativement. La comparaison de ces deux figures illustre l'invariance du taux NP pour les algorithmes ID-MEFPA et ID-PPPA et la dégradation de ce taux pour l'algorithme kpID-MCP lorsque les poids sont corrélés négativement.

Le coût MC

Les figures 33, 34 et 35 illustrent la variation du coût MC en fonction des contraintes de qualité de service dans la topologie LatticeSL(2,3) respectivement avec un nombre de contraintes $k = 2$, $k = 3$ et $k = 4$. La comparaison des trois figures montre que le coût MC reste invariant pour l'algorithme kpID-MCP et diminue légèrement pour les algorithmes ID-PPPA et ID-MEFPA lorsque le nombre de contraintes k augmente. Ceci, illustre l'effet de l'augmentation du nombre de contraintes sur la qualité des chemins retournés par les algorithmes ID-MEFPA et ID-PPPA.

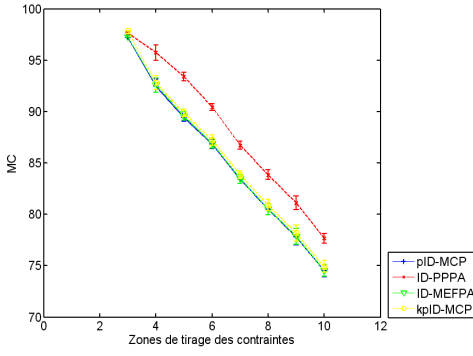


FIGURE 33 – MC en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids non corrélés

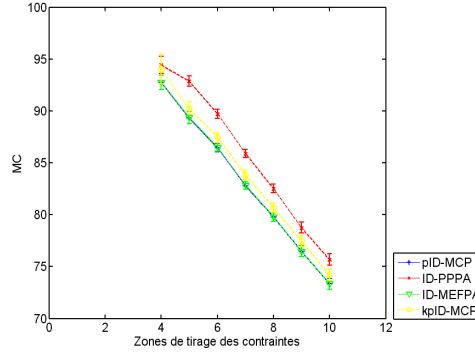


FIGURE 34 – MC en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 3$ et des poids non corrélés

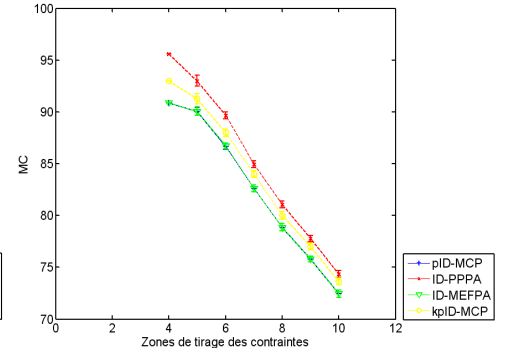


FIGURE 35 – MC en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

Le coût C

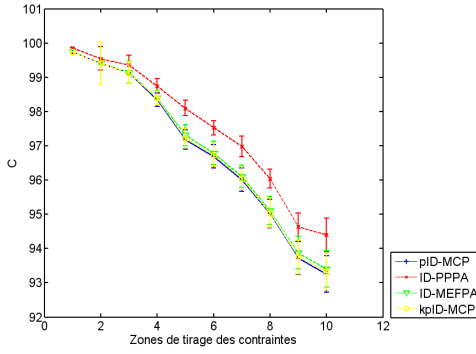


FIGURE 36 – C en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés positivement

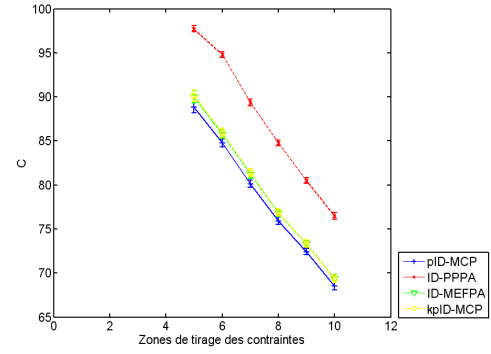


FIGURE 37 – C en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 2$ et des poids corrélés négativement

Les figures 36 et 37 illustrent la variation du coût C en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) respectivement avec des poids corrélés positivement et des poids corrélés négativement. La comparaison de ces deux figures montre que le coût C est plus faible lorsque les poids sont corrélés négativement. Ceci illustre le fait que la qualité des chemins retournés est meilleure lorsque les poids sont corrélés négativement, comme expliqué dans 4.2.3.

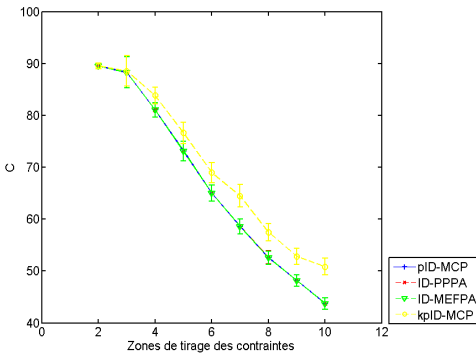


FIGURE 38 – C en fonction des contraintes de qualité de service dans la topologie LatticeFM(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

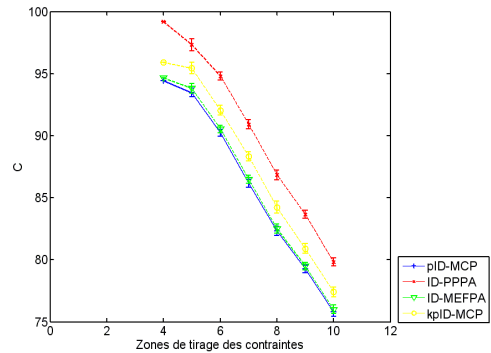


FIGURE 39 – C en fonction des contraintes de qualité de service dans la topologie LatticeSL(25,3) avec un nombre de contraintes $k = 4$ et des poids non corrélés

Les figures 38 et 39 illustrent la variation du coût C en fonction des contraintes de qualité de service respectivement dans les topologies LatticeFM(25,3) et LatticeSL(25,3) avec un nombre de contraintes $k = 4$. La comparaison de ces deux figures illustre l'effet de connectivité inter-domaine sur la qualité des chemins retournés. En effet, le coût C mesuré en utilisant la topologie LatticeFM(25,3) est largement inférieur à celui mesuré en utilisant la topologie LatticeSL(25,3).