Spatial Dantzig-Wolfe decomposition for multi-commodity flow problem

Youcef Magnouche
Huawei Technologies Ltd.,
Paris Research Center, France
youcef.magnouche@huawei.com

Sebastien Martin
Huawei Technologies Ltd.,
Paris Research Center, France
sebastien.martin@huawei.com

***Abstract*—Solving the multi-commodity flow problem in large networks is the core of current telecommunication networks. Integer linear programming shows its ability to solve this problem efficiently in large networks by using decomposition techniques such as the Dantzig-Wolfe method to obtain a model based on path variables. Thanks to the sparsity of telecommunication networks, the path model can solve large-scale instances through a small mathematical model. This paper proposes a spatial decomposition derived from the Dantzig-Wolfe decomposition method. The goal is to split the network into several areas and treat each one independently. Column generation is used to find a routing consensus between areas. We show the gain in terms of linear relaxation and computational time using spatial decomposition on the well-known IPRAN network structure.**

***Index Terms*—Telecommunication, Routing, Optimization, Column Generation, Decomposition, Multi-commodity flow problem.**

I. INTRODUCTION

The multi-commodity flow problem is a network flow problem where multiple distinct commodities are transmitted from several sources to several destinations through a shared network. Commodities may have different source and/or sink nodes and are routed over the same network with limited link bandwidth capacities. The goal is to find a feasible flow for each commodity that satisfies the link capacity constraints while optimizing a specified objective, such as minimizing the maximum link utilization.

The multi-commodity flow problem has many applications in telecommunications. With the emergence of 5G, many new constraints have appeared, such as Slicing [13], DetNet [9], DIP [1, 14, 15], QoS [11]. These new constraints make the multi-commodity problem harder to solve. Moreover, the network sizes and the number of services are growing, which requires efficient algorithms to tackle the multi-commodity flow problem.

In the following, we consider the multi-commodity flow problem where the goal is to find a set of paths for each commodity to route the traffic demand such that the Maximum Link Utilization (MLU) is minimized. In the unsplittable version, only one path is computed per commodity, otherwise it is called splittable. Formally, K is a set of commodities where each commodity $k \in K$ is represented by a triplet: source, destination, and bandwidth demand in Mbps, denoted (s_k, t_k, b_k), respectively. Let us consider a graph $G = (V, A)$ where V represents the set of vertices and A is the set of arcs. Let $c_a \in \mathbb{R}^+$ be the link capacity of the arc $a \in A$ in Mbps. We denote by $\delta^+(v)$ (resp. $\delta^-(v)$) the set of outgoing (resp. ingoing) arcs of vertex $v \in V$.

The basic mathematical model to solve the unsplittable multi-commodity flow problem, considers two families of variables: MLU that represents the maximum link utilization in percentage, and x_a^k that equals 1 if the commodity k crosses link a, 0 otherwise. This model is called the arc flow model [16]:

$$\min \quad MLU \tag{1}$$

$$\sum_{a\in\delta^+(v)} x_a^k - \sum_{a\in\delta^-(v)} x_a^k = \begin{cases} 1 \text{ if } v = s^k, \\ -1 \text{ if } v = t^k, \forall k \in K, \forall v \in V, \\ 0 \text{ otherwise}; \end{cases} \tag{2}$$

$$\sum_{k\in K} b_k \sum_{k\in P_a^k} x_a^k \leq c_a MLU \qquad \forall a \in A, \tag{3}$$

$$x_a^k \in \{0,1\} \qquad \forall k \in K, \forall a \in A, \tag{4}$$

where (2) are the flow conservation constraints and (3) are the capacity constraints where the MLU variable appears. To solve the splittable version of the multi-commodity flow it is enough to replace integrality constraints (4) by $0 \leq x_a^k \leq 1$ for all $a \in A$ and all $k \in K$.

The major drawback of this model is the number of variables and constraints, where variables represent the utilization of a link by a commodity. An extended model obtained thanks to the Dantzig-Wolfe decomposition, called the path model, provides better scalability to get the linear relaxation. In this model, each variable represents a path [16], which requires a column generation algorithm to get the linear relaxation, where the pricing consists of solving the shortest path problem. To obtain an integer solution, i.e., imposing exactly one path per commodity, a branch-and-price algorithm [8] must be developed to get the optimal solution, or a metaheuristic [7] to derive a heuristic solution. Another Dantzig-Wolfe decomposition-based model is considered in the literature, called the pattern model [6], where the variables correspond to a set of commodities for each link. Other decompositions are considered for flow problems in [12] and [2] where the goal is to improve linear relaxation by adding hard constraints in pricing problems.

The main advantage of the path model is the time it needs to reach the linear relaxation, whereas the advantage of the pattern model is on the quality of the linear relaxation. Our main contribution, in this paper, consists in providing a new Dantzig-Wolfe decomposition exploiting the network topology.

Spatial decomposition allows us to decompose the main problem into multiple subproblems based on the problem characteristics like topologies. Several researchers have investigated this decomposition in different fields, such as energy [4, 10] and facility location [5]. In this paper, we consider another way to decompose the problem based on graph partitioning. In Section 2, we provide the new decomposition in which the network is divided into several areas. We focus on a specific case, IPRAN networks, leading to a new model called the partial spatial decomposition model. Section 3 presents some improvements of column generation: the pricing filtering and warm start. Finally, Section 4 presents the experimental results to show the efficiency of our approach in different instances.

II. SPATIAL DECOMPOSITION FOR MULTI-COMMODITY FLOW PROBLEM

In this section, we first provide a generic model, valid for any spatial decomposition. Consider a given network decomposition I, where for each area $i \in I$, V_i is the set of associated nodes. Let us denote by $\mathcal{V} = \{V_0, ..., V_{|I|}\}$ the set of all areas' nodes. For $v \in V$, let $\mathcal{V}_v \subseteq \mathcal{V}$ be the subset that contains node v. Let $\bar{V} = \bigcup_{i,j \in I | i \neq j} (V_i \cap V_j)$ be the set of nodes between two areas. It is easy to see that if for a node $v \in V$, $|\mathcal{V}_v| \geq 2$ then $v \in \bar{V}$.
Let us denote by $H_i = (V_i, A_i)$ the graph associated with area $i \in I$. Graph H_i contains additional virtual nodes, $\bar{v}_j$ for every area $j \in I \setminus \{i\}$ where $|V_i \cap V_j| > 0$. This vertex $\bar{v}_j$ is connected to every vertex of $V_i \cap V_j$ with an infinite capacity arc. For example, Figure 1 shows a network and Figure 2 illustrates one possible decomposition where virtual vertices are added $\bar{v}_j$.

a) Variables: For spatial decomposition, we consider an exponential number of variables, where each of them represents the routing of some commodities in one area. In [2] the authors propose a similar idea where variables represent the routing on the entire graph. For $i \in I$, we denote by R_i all possible routings in area i.

A routing $r \in R_i$ in an area $i \in I$ is restricted to the graph H_i and represents a set of paths for each commodity. Remark that for a given commodity, the path can be empty. Let us denote by $\beta_r^{i+}(v,k)$ (resp. $\beta_r^{i-}(v,k)$) a value equal to 1 if the path of commodity k goes into (resp. leaves) area i by crossing node v, 0 otherwise.

We define the following variables:

- MLU: the maximum link utilization,
- x_r^i: equals to 1 if the routing $r \in R_i$ is applied to the area $i \in I$, 0 otherwise.

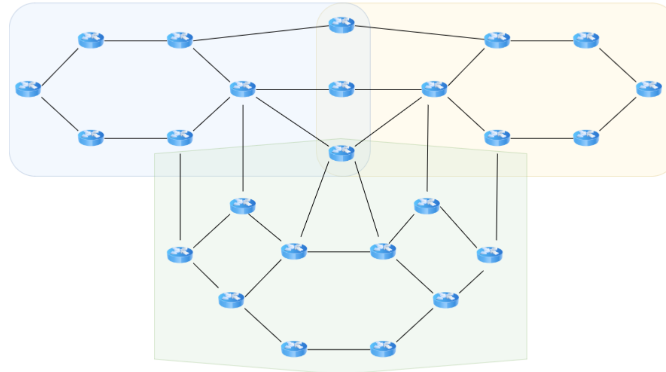

Fig. 1. Example of a telecommunication network with three identified areas

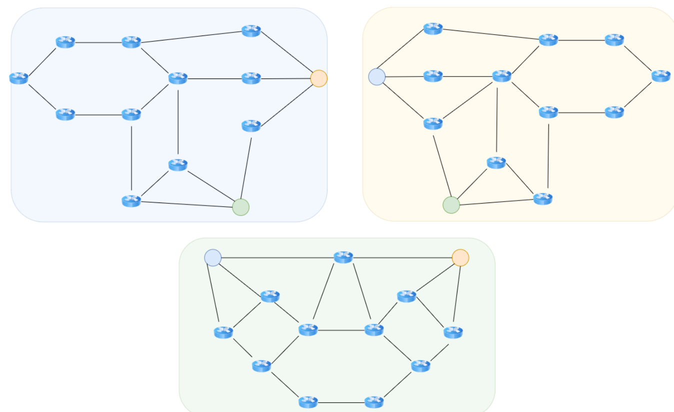

Fig. 2. Example of three areas after decomposition. Virtual nodes are added.

b) Mathematical Model:

$$\min \quad MLU \tag{5}$$

$$\alpha_i : \sum_{r \in R_i} x_r^i = 1 \qquad \forall i \in I \cup \{0\}, \tag{6}$$

$$\alpha_{MLU}^i : MLU - \sum_{r \in R_i} MLU_r^i x_r^i \geq 0 \quad \forall i \in I \cup \{0\}, \tag{7}$$

$$\alpha_{v,k} : \sum_{V_i \in \mathcal{V}_v} \sum_{r \in R_i} \beta_r^{i+}(v,k) x_r^i - \sum_{V_i \in \mathcal{V}_v} \sum_{r \in R_i} \beta_r^{i-}(v,k) x_r^i = 0 \quad \forall v \in \bar{V}, \forall k \in K, \tag{8}$$

$$x_r^i \in \{0,1\} \qquad i \in I, r \in R_i. \tag{9}$$

where (6) are the convexity constraints, (7) are related to the capacity constraints, and (8) are the flow conservation constraints between two areas. The variables α_i, α_{MLU}^i, and $\alpha_{v,k}$ are dual variables associated with (6), (7), and (8),

Remark that the flow conservation constraints do not consider a right-hand-side to 1 or -1. The source and destination of the commodities are managed in the pricing.

This model has an exponential number of variables. To solve this model, we first have to solve the linear relaxation of the model, by replacing integrity constraints (9) by $x_r^i \geq 0$ for all

$i \in I$ and $r \in R_i$. Let us focus on the associated pricing for dynamically generating variables.

c) Pricing: For each $i \in I$, the pricing problem consists of finding a routing r^* in the graph H_i where $\alpha_i - \alpha^i_{MLU} MLU^i_{r^*} + \sum_{v\in V}\sum_{k\in K}\beta^{i+}_r(v,k)\alpha_{v,k} - \sum_{v\in V}\sum_{k\in K}\beta^{i-}_r(v,k)\alpha_{v,k} > 0$. If such routing exists, then a column is added. That can be solved with the following ILP model.

$$\begin{aligned}\min \quad & \alpha^i_{MLU} MLU \\ & + \sum_{j\in I:|V_i\cap V_j|>0}\ \sum_{(\bar{v}_j,v)\in\delta^+(\bar{v}_j)}\ \sum_{k\in K}\alpha_{v,k}(x^k_{(\bar{v}_j,v)}) \\ & - \sum_{j\in I:|V_i\cap V_j|>0}\ \sum_{(v,\bar{v}_j)\in\delta^-(\bar{v}_j)}\ \sum_{k\in K}\alpha_{v,k}(x^k_{(v,\bar{v}_j)})\end{aligned}$$

$$\sum_{a\in\delta^+(v)} x^k_a - \sum_{a\in\delta^-(v)} x^k_a = \begin{cases}1 \text{ if } v = s^k, \\ -1 \text{ if } v = t^k, \forall k\in K, \forall v\in V_i, \\ 0 \text{ otherwise;}\end{cases} \quad (10)$$

$$\sum_{j\in I:|V_i\cap V_j|>0}\ \sum_{a\in\delta^+(\bar{v}_j)} x^k_a - \sum_{j\in I:|V_i\cap V_j|>0}\ \sum_{a\in\delta^-(\bar{v}_j)} x^k_a = 0 \qquad \forall k\in K, \forall j\in I\setminus\{0\}, \quad (11)$$

$$c_a MLU - \sum_{k\in K} b_k \sum_{k\in P^k_a} x^k_a \geq 0 \qquad \forall a\in A, \quad (12)$$

$$x^k_a \in \{0,1\} \qquad \forall k\in K, a\in A, \quad (13)$$

where (10) and (11) are the flow conservation constraints, and inequalities (12) are the capacity constraints.

Pain point of this approach: The main pain point comes from how to decompose the graph to get an efficient model. It is important to load balance the difficulty of solving each set of the decomposition. This difficulty comes from the number of nodes, the density and also the average time where a commodity needs to cross this area.

In the following, we focus on a well-known structure of telecommunication networks, called IPRAN. Using our approach in this kind of network allows proposing a decomposition by design.

A. Spatial decomposition for IPRAN Networks

IPRAN (IP Radio Access Network) is a network architecture used in telecommunications. They are designed to improve the efficiency, scalability, and flexibility of mobile networks by allowing the integration of multiple types of services and applications over a unified IP-based infrastructure. IPRAN networks are composed of a core network that connects different aggregated networks. For some IPRAN networks, a third layer is considered where the aggregated network connects different access networks. In the rest of this paper, we focus on a network with one core network and several aggregated networks. Due to redundancy, two areas are connected with two nodes or not connected. For some specific cases, only one vertex can connect two areas. This particular case allows us to split the problem into two independent sub-problems. The main characteristic comes from the fact that removing the core network disconnects all aggregated networks.

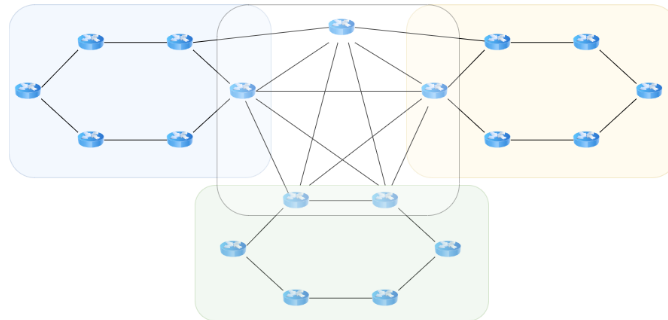

Fig. 3. Example of IPRAN network. The core network is in white colour, connected to three aggregated networks.

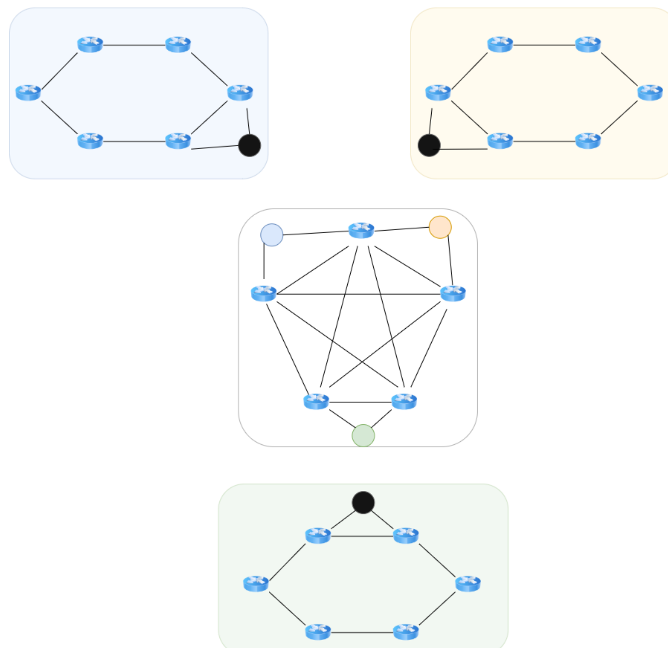

Fig. 4. Example of IPRAN network decomposition

In Figure 3, an example of an IPRAN network. The core network connects three access networks. Figure 4 shows a possible decomposition of this network.

We consider the following decomposition $\mathcal{V} = \{V_0, ..., V_{|I|}\}$ where V_0 represents the core, and thus removing V_0 disconnects all other sets and V_i represents the aggregation networks for all $i \in I \setminus \{0\}$.

We consider the same family of variables described in the previous section. In the preliminary test, this model needs a lot of time to converge. For this reason, we propose another model where the spatial decomposition and the compact model are merged to improve linear relaxation and convergence.

B. Partial Spatial Decomposition

In this formulation, we consider the compact model variables for the core network and the spatial decomposition model variables for the other areas. We obtain the following mathematical model.

$$\min \quad MLU \tag{14}$$

$$\alpha_i : \sum_{r\in R_i} x^i_r = 1 \qquad \forall i \in I \setminus \{0\}, \tag{15}$$

$$\alpha^i_{MLU} : MLU - \sum_{r\in R_i} MLU^i_r x^i_r \geq 0 \qquad \forall i \in I \setminus \{0\}, \tag{16}$$

$$\sum_{k\in K} b_k \sum_{k\in P^k_a} x^k_a \leq c_a MLU \qquad \forall a \in A_0, \tag{17}$$

$$\begin{aligned}\alpha_{v,k} : &\sum_{k\in K} \sum_{a\in A_0\cap(\delta^+(v))} x^k_a \\ &- \sum_{k\in K} \sum_{a\in A_0\cap(\delta^-(v))} x^k_a \\ &- \sum_{V_i\in \mathcal{V}_v} \sum_{r\in R_i} \beta^{i+}_r(v,k) x^i_r \\ &+ \sum_{V_i\in \mathcal{V}_v} \sum_{r\in R_i} \beta^{i-}_r(v,k) x^i_r = \begin{cases} 1 \text{ if } v = s^k, \\ -1 \text{ if } v = t^k, \\ 0 \text{ otherwise;} \end{cases} \\ &\forall i \in I \setminus \{0\}, \forall v \in V_i \cap V_0 \forall k \in K, \end{aligned} \tag{18}$$

$$x^i_r \in \{0,1\} \qquad i \in I \setminus \{0\}, r \in R_i, \tag{19}$$

$$x^k_a \in \{0,1\} \qquad \forall k \in K, \forall a \in A. \tag{20}$$

To solve this model, the same pricing problems as presented in the previous section can be used.

III. Improvements

In this section, we propose three techniques to speed-up the column generation algorithm.

A. Case: polynomial pricing

Proposition 1. *If the dual variable α^i_{MLU} is equal to 0 for a given area $i \in I$ then the associated pricing problem can be solved in polynomial time.*

Proof. It is easy to see that if $\alpha^i_{MLU} = 0$, the capacity constraints can be ignored. This implies that the problem can be solved for each commodity. Therefore, consider a commodity $k \in K$, and note that some links have a positive or negative cost. To provide a polynomial algorithm, we need to remove negative costs. The cost of links in the pricing problem is only related to the links ingoing or outgoing node $\bar{v}_j$. The commodity k must cross a node $\bar{v}_j$ exactly once. Thus, we can subtract the smallest negative cost of ingoing (resp. outgoing) link of $\bar{v}_j$ to compute the shortest path and replace the cost of the path with the original dual cost. This implies a polynomial algorithm to solve the pricing problem associated with $i \in I$ when $\alpha^i_{MLU} = 0$. □

B. Pricing filtering

Column generation is an iterative algorithm, where in each iteration, one master problem and multiple pricing problems are solved. Solving pricing problems represents the major part of the whole column generation solving time. Furthermore, solving one pricing problem does not guarantee the generation of one column, causing a waste of time. In [3], authors propose an interesting way of skipping solving some pricing problems if they cannot generate columns at one iteration. This is based only on the previous dual values. In the following, we prove a similar checking way for our Spatial decomposition problem.
For area $i > 0$. Consider an iteration $l < t$. The minimum reduced cost is given by

$$\bar{c}^i_t = -\alpha^t_i + \min_{r\in R}(\alpha^{it}_{MLU} MLU^i_{r^*} + \sum_{v\in V_i\cap V_0} \sum_{k\in K_i} \alpha^t_{i,v,k})$$

This is equivalent to

$$\begin{aligned}\bar{c}^i_t = -\alpha^t_i + \min_{r\in R}(\alpha^{it}_{MLU} MLU^i_{r^*} + \sum_{v\in V_i\cap V_0} \sum_{k\in K_i} \alpha^t_{i,v,k}) + \alpha^l_i \\ - \min_{r\in R}(\alpha^{il}_{MLU} MLU^i_{r^*} + \sum_{v\in V_i\cap V_0} \sum_{k\in K_i} \alpha^l_{i,v,k})\end{aligned}$$

Then

$$\begin{aligned}\bar{c}^i_t \geq \alpha^l_i - \alpha^t_i + \min_{r\in R}((\alpha^{it}_{MLU} - \alpha^{il}_{MLU}) MLU^i_r \\ + \sum_{v\in V_i\cap V_0} \sum_{k\in K_i} (\alpha^t_{i,v,k} - \alpha^l_{i,v,k}))\end{aligned}$$

When $MLU^i_r \geq 0$, we suppose that MLU^i_r equals 0, otherwise, it equals the highest value that MLU^i_r can take, given from a heuristic for example. Therefore, if the inequality is respected, the associated pricing problem cannot give a column with a negative reduced cost.

C. Warm Start

Warm start allows column generation to start with interesting columns. To warm-start our column generation, we propose two heuristics. The first is a greedy heuristic that quickly provides a column for each area. The second is based on a mathematical model that provides efficient columns for all areas.

1) Greedy Heuristic: The following greedy heuristic consists in finding a feasible solution on the entire graph G and deducing the associated column for each area. To compute the solution, we sequentially compute the shortest path for each commodity where the cost is modified to consider the MLU at each iteration.

2) Matheuristic: The second heuristic consists of solving the multi-commodity flow problem in each area to get the best local MLU. These columns allow for finding the areas where the MLU is higher, and the column generation will first focus on these areas.

IV. Experimental results

In this paper, we evaluate our algorithms using a machine with Intel(R) Xeon(R) CPU E5-4627 v2of 3.30GHz with 504GB RAM, running under Linux 64 bits OS. The time limit is set to 600 seconds, using one thread. The set of

Algorithm 1 Greedy Heuristic

Input:
Network $G = (V, A)$
Demands K, Areas I
Output: Set of columns for each area
$u_a = 0$ for all $a \in A$ (link utilization)
for $k \in K$ **do**
  $w_a = \frac{u_a + b_k}{c_a}$ for each $a \in A$
  compute the shortest path p_k from s_k to d_k with the cost w
  $u_a = u_a + b_k$ for all $a \in p_k$
end for
$r_i = \cup_{k \in K}(p_k \cap H_i)$ for all $i \in I$
Return routing r_i for all $i \in I$

Algorithm 2 Matheuristic

Input:
Network $G = (V, A)$
Demands D, Areas I
Output: Set of columns for each area
for $i \in I$ **do**
  Solve compact model $(1) - (4)$ in area i
end for
Return, for each area, the associated routing

instances is generated randomly. The latter has been realized in such a way as to respect the IPRAN structure, i.e., 1 core network with multiple Aggregated areas connected to the core network. Commodities are generated with random sources/destinations in aggregated areas and random traffic demands. Finally, link capacities are also randomly defined.
Core network represents the bottleneck for the multi-commodity flow problem. Indeed, all traffic between two different areas must spend through the core network. In practice, links in the core network have much more capacity than the aggregated areas. In this paper, we analyze the impact of this bottleneck by multiplying the link capacities of the core network by a parameter $\alpha \in \{1, 5, 10, 20\}$ (when $\alpha = 1$, the core network has the same magnitude of capacity as the aggregated areas). Similar to the link capacities of the core network, this later has a much higher density compared to the aggregated area. In this paper, we suppose that the core network is a complete graph and varies the number and size of aggregated areas together with the associated density, through parameters $aggNb \in \{3, 5, 10\}$, $aggSize \in \{10, 25, 50\}$ and $aggDensity \in \{0.1, 0.25, 0.5\}$, respectively.

For a complete analysis of the problem, we consider the following variants:

- Compact: corresponds to model $(1) - (4)$,
- CG: corresponds to model $(5) - (7)$,
- CompactRL: corresponds to linear relaxation of Compact,
- CGRL: corresponds to CG when pricing problems $(14) - (20)$ are linearly relaxed.

We also show the impact of the pricing filtering. Therefore, we solve two versions of CG and CGRL, with filtering "CG_with"-"CGRL_with" and without filtering "CG_without"-"CGRL_without".

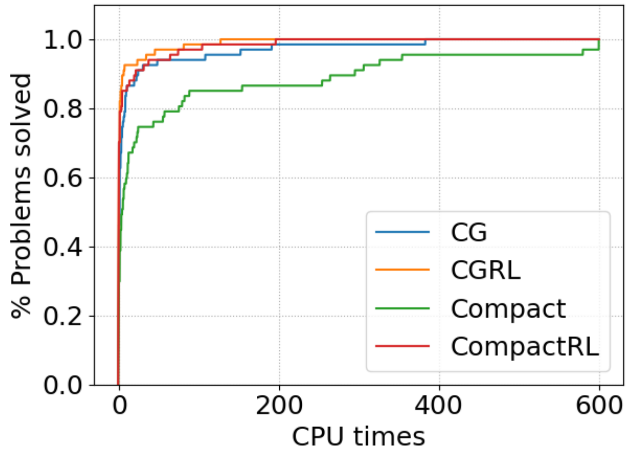

Fig. 5. Performance profile comparing the CPU times of all variants

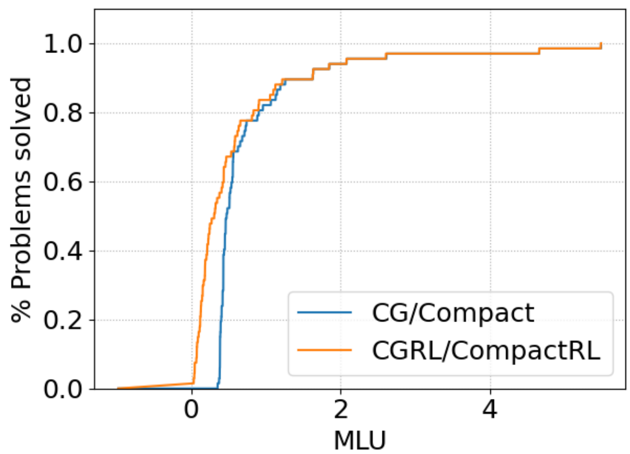

Fig. 6. CPU time comparison when capacities of Core network is increased by 20 times

Figure 5 compares the CPU time of the four variants. We notice that the relaxed versions are the fastest. Except for Compact, all other variants solve 80% of instances in a few seconds and solve all instances in less than 400 seconds. In contrast, the Compact variant is the slowest one, with some instances reaching the time limit. Figure 6 compares the MLU value of the four variants. As expected, we notice that when traffic can be split over multiple paths, a lower MLU can be found. Moreover, the integer version (resp. relaxed versions) gives the same MLU. An interesting aspect comes from the quality of the column generation. The CG algorithm found the optimal solution in all our instances. It is not needed to branch to obtain an integer solution. This is due to the integrity of the pricing problems. Figure 7 compares the number of columns generated by the column generation algorithm in the CG version, with and without pricing filtering. Clearly, we notice that the filtering allows us to decrease the number of

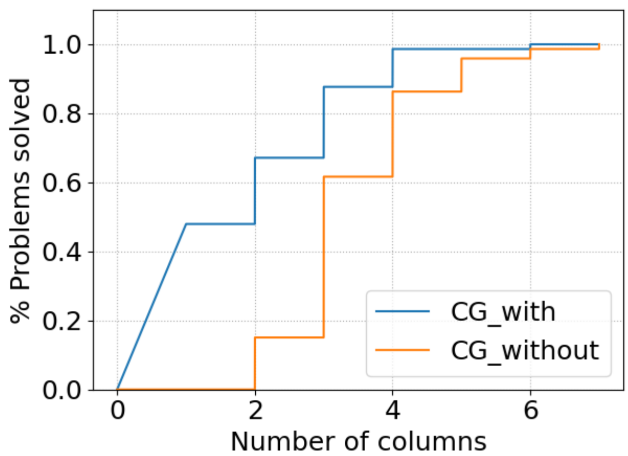

Fig. 7. Comparison of the number of Columns in the columns generation algorithm with and without filtering the pricing problems

columns.

V. Conclusion and perspectives

In this paper, we have proposed a spatial decomposition based on the Dantzig-Wolfe method for the multi-commodity flow problem that enhances algorithm-solving performance. Combining the Dantzig-Wolfe decomposition and the topology structure, we demonstrated the efficiency of the proposed algorithm. Future work in this area can address more complex scenarios where the topology is not easily decomposable, which requires solving a partitioning algorithm to define several regions. Moreover, this work can be extended to other optimization problems to achieve solving a large-scale problem.

References

[1] Vincent Angilella, Filip Krasniqi, Paolo Medagliani, Sébastien Martin, Jérémie Leguay, Ren Shoushou, and Liu Xuan. High capacity and resilient large-scale deterministic ip networks. *J Netw Syst Manage*, 30:1573–7705, 2022.

[2] Amal Benhamiche, Morgan Chopin, and Sébastien Martin. Unsplittable shortest path routing: Extended model and matheuristic. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 926–931, 2023.

[3] Abdellah Bulaich Mehamdi, Mathieu Lacroix, and Sébastien Martin. Pricing filtering in dantzig-wolfe decomposition. *Operations Research Letters*, 58:107207, 2025.

[4] Minas Chatzos, Terrence W. K. Mak, and Pascal Van Hentenryck. Spatial network decomposition for fast and scalable ac-opf learning. *IEEE Transactions on Power Systems*, 37(4):2601–2612, 2022.

[5] Clarisse Dhaenens-Flipo. Spatial decomposition for a multi-facility production and distribution problem. *International Journal of Production Economics*, 64(1):177–186, 2000.

[6] Zhang Fan, Wang Jiazheng, Mathieu Lacroix, Roberto Wolfler Calvo, Youcef Magnouche, and Sebastien Martin. The multi-commodity flow problem: Double dantzig-wolfe decomposition. In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1171–1176, 2024.

[7] Martina Fischetti and Matteo Fischetti. *Matheuristics*, pages 121–153. Springer International Publishing, Cham, 2018.

[8] M. Gamst and B. Petersen. Comparing branch-and-price algorithms for the multi-commodity k-splittable maximum flow problem. *European Journal of Operational Research*, 217(2):278–286, 2012.

[9] Jonatan Krolikowski, Sébastien Martin, Paolo Medagliani, Jérémie Leguay, Shuang Chen, Xiaodong Chang, and Xuesong Geng. Joint routing and scheduling for large-scale deterministic ip networks. *Computer Communications*, 165:33–42, 2021.

[10] Arjun Kumar Madhusoodhanan, Marco Giuntoli, Susanne Schmitt, Iiro Harjunkoski, Jan Poland, and Thomas Leibfried. A spatial decomposition method for solving the security-constrained unit commitment problem. In *2024 IEEE International Humanitarian Technologies Conference (IHTC)*, pages 1–7, 2024.

[11] Youcef Magnouche, Pham Tran Anh Quang, Jérémie Leguay, Xu Gong, and Feng Zeng. Distributed load balancing from the edge in ip networks. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

[12] Sebastien Martin, Pierre Bauguion, Youcef Magnouche, and Jérémie Leguay. Atomic column generation for consensus between algorithms: Application to path computation. *Networks*, 2025.

[13] Sebastien Martin, Paolo Medagliani, and Jeremie Leguay. Network slicing for deterministic latency. In *2021 17th International Conference on Network and Service Management (CNSM)*, pages 572–577, 2021.

[14] M. Yassine Naghmouchi, Shoushou Ren, Paolo Medagliani, Sébastien Martin, and Jérémie Leguay. Scalable damper-based deterministic networking. In *2022 18th International Conference on Network and Service Management (CNSM)*, pages 367–373, 2022.

[15] M. Yassine Naghmouchi, Shoushou Ren, Paolo Medagliani, Sébastien Martin, and Jérémie Leguay. Optimal admission control in damper-based networks: Branch-and-price algorithm. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 488–493, 2023.

[16] Khodakaram Salimifard and Sara Bigharaz. The multicommodity network flow problem: state of the art classification, applications, and solution methods. *Operational Research*, 22(1):1–47, Mar 2022.