Improving Dual Bounds for the Unsplittable Multicommodity Capacitated Network Design Problem

Lacy M. Greening*[,1], Santanu S. Dey[2], Alan L. Erera[2]

[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ

[2]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA

*Corresponding author (lacy.greening@asu.edu)

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)

June 22-27, 2025, Okinawa, Japan

February 14, 2025

Keywords: network design; valid inequalities; fulfillment logistics;

1 Introduction

The rapid growth of e-commerce has highlighted the importance of efficient fulfillment logistics in managing costs and ensuring on-time delivery. Central to the design of e-commerce fulfillment networks is the *multicommodity capacitated network design* (MCND) problem, which seeks to optimize the shipment of commodities—defined as origin-destination pairs with specific demand volumes—through a transshipment network at minimum cost. In this context, shipping demand between a pair of locations requires installing capacity (e.g., trailers) on directed arcs, incurring a fixed cost for each unit installed, and possibly a variable cost based on demand volume transported.

Real-world fulfillment networks require unsplittable (or non-bifurcated) flows when deciding shipment paths through the network. Although it simplifies operations, this requirement can make it even more challenging to solve realistically-sized instances optimally, as it necessitates converting continuous flow variables into binary variables. Numerous works focus on developing effective metaheuristic approaches to identify strong primal solutions for integer programming (IP) models of logistics and transportation MCND problems (Crainic & Gendreau, 2021); however, assessing the quality of these solutions is often difficult for realistically-sized instances due to weak dual bounds—a common issue in network design problems caused by their weak linear programming (LP) relaxations. Thus, this talk will focus on strengthening the LP relaxation of the MCND formulation by introducing a new class of valid inequalities for the unsplittable MCND problem (applicable to both arc- and path-based formulations), highlighting their potential applicability to sophisticated solution algorithms.

2 Formulation

The MCND problem is to determine a minimum-cost allocation of transportation capacity on network arcs to ensure that commodity demand requirements are satisfied. Thus, let $G = (\mathcal{N}, \mathcal{A})$ define a transshipment network, where $\mathcal{N}$ represents a set of locations (nodes) in the network and $\mathcal{A}$ represents a set of directed arcs connecting pairs of locations. Shipment demand is modeled using a set $\mathcal{K}$ of commodities, where each commodity $k \in \mathcal{K}$ has an origin $o(k) \in \mathcal{N}$, destination $d(k) \in \mathcal{N}$, and demand volume d_k that must be shipped from origin to destination along a single path (or sequence of adjoined arcs) through the network. Let c_{ij} represent the variable cost of transporting one unit of demand on arc $(i, j) \in \mathcal{A}$, f_{ij} represent the fixed cost of activating (i.e., installing one unit of capacity) on arc $(i, j) \in \mathcal{A}$, and let q_{ij} represent the capacity of arc $(i, j) \in \mathcal{A}$. Let $x^k_{ij} \in \{0, 1\}$ be a binary variable indicating if commodity $k \in \mathcal{K}$ is transported via arc $(i,j) \in \mathcal{A}$ or not. Let $y_{ij} \in \{0,1\}$ be a binary variable indicating if arc $(i,j) \in \mathcal{A}$ is activated to transport commodity demands or not. The arc-based unsplittable MCND problem is:

$$\min_{x,y} \quad \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c_{ij} d_k x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} \tag{1a}$$

$$\text{s.t.} \quad \sum_{(i,j)\in\delta^+(i)} x_{ij}^k - \sum_{(j,i)\in\delta^-(i)} x_{ji}^k = \begin{cases} 1, & i = o(k), \\ 0, & i \neq o(k), d(k), \\ -1, & i = d(k), \end{cases} \qquad \forall\, i \in \mathcal{N},\ \forall\, k \in \mathcal{K}, \tag{1b}$$

$$\sum_{k\in\mathcal{K}} d_k x_{ij}^k \leq q_{ij} y_{ij}, \qquad \forall\,(i,j) \in \mathcal{A}, \tag{1c}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall\,(i,j) \in \mathcal{A},\ \forall\, k \in \mathcal{K}, \tag{1d}$$

$$y_{ij} \in \{0,1\}, \qquad \forall\,(i,j) \in \mathcal{A}, \tag{1e}$$

where $\delta^+(i)$ and $\delta^-(i)$ refer to the sets of arcs emanating from and ending at node $i \in \mathcal{N}$, respectively. Constraints (1b) ensure flow balance. Constraints (1c) activate arc $(i,j) \in \mathcal{A}$ and ensure its capacity is not exceeded.

3 Single-Arc Commodity Packing Valid Inequalities

Strong valid inequalities can potentially be generated from simple structured relaxations over more complicated sets, like those focused on the unsplittable flow arc-set polyhedron (i.e., the convex hull of solutions to capacity constraints of the form (1c)) (Atamtürk & Rajan, 2002, Chen *et al.*, 2021, Bienstock *et al.*, 1998). Thus, we investigate a simple structured relaxation of (1) in which we also focus on the capacity-related constraints.

For a given arc $a \in \mathcal{A}$, we consider the following set:

$$S := \left\{ (x,y) \in \{0,1\}^{|\mathcal{K}|} \times \{0,1\} \mid \sum_{k\in\mathcal{K}} d_k x_a^k \leq q_a y_a \right\}. \tag{2}$$

We first analyze the convex hull solutions of (2), from which we deduce the general form of the constraints defining the set.

Proposition 1. *There exists non-negative matrices G, H such that*

$$\operatorname{conv}(S) = \left\{ (x,y) \in [0,\ 1]^{|\mathcal{K}|} \times [0,\ 1] \mid Gx \leq Hy \right\}.$$

It is important to note that the general form of the constraints lacks a constant term in the constraint $Gx \leq Hy$, unlike the inequality introduced in Chen *et al.* (2021). The reason for this is arcs do not have pre-existing capacities, which would introduce a constant term to the right-hand side of (1c). We now describe new valid inequalities for S that are motivated by Proposition 1.

Since G is a non-negative matrix, we first explore a subset of constraints which involve only $\{0,1\}$ coefficients for the arc variables x. In this case, given both x and y are binary variables, we know that each element in H is bounded above by $|\mathcal{K}|$. Observe that for a given a left-hand-side binary vector g corresponding to the x-vector, we can explicitly determine h, the coefficient for y, by counting the maximum number of *commodities* selected in g (i.e., x_k variable with a coefficient of 1 in g) that can be transported using (or *packed* into) q_a capacity units. We state this result formally next.

Proposition 2. *Consider a non-zero binary vector g. The inequality*

$$\sum_{k\in\mathcal{K}} g_k x_k \leq \alpha y, \tag{3}$$

is a valid inequality for S if

$$\alpha = \max \left\{ \sum_{k \in \mathcal{Z}} x_k \;\middle|\; \sum_{k \in \mathcal{Z}} d_k x_k \leq q_a, x_k \in \{0,1\} \right\}, \tag{4}$$

where $\mathcal{Z} = \{k \in \mathcal{K} \,|\, g_k = 1\}$.

Using Proposition 2, we define the *single-arc commodity packing* (SAC-Pack) constraints as:

$$\sum_{k \in \mathcal{Z}} x_a^k \leq \alpha_a y_a, \quad \forall\, a \in \mathcal{A}, \tag{5}$$

where α_a is an integer coefficient equal to the maximum number of selected commodities that can be transported by q_a capacity units. In a sense, constraints (5) are a "smart re-aggregation" of the typical MCND linking constraints for a set of commodities transported via arc a, and can at times even dominate them.

Separation. Given the set S described in (2) and a point (x^*, y^*), there exists a SAC-Pack constraint (5) violated by (x^*, y^*) if and only if there exists a set $\mathcal{Z} \subseteq \mathcal{K}$ such that $\sum_{k \in \mathcal{Z}} x_k^* - \alpha y^* > 0$. Because we can determine the α value in closed form (see Proposition 2), we can explicitly check for such a violation by formulating the problem as an IP with the objective $\max_{\mathcal{Z} \subseteq \mathcal{K}} \{\sum_{k \in \mathcal{K}} z_k x_k^* - \alpha y^*\}$, where z_k is a binary variable indicating whether commodity k is selected to be in $\mathcal{Z}$, and α is an integer variable representing the number of selected commodities that can be transported by q_a capacity units. If the objective value is greater than 0, we add the violated constraint to the model.

4 Results

To assess the strength of our new valid inequalities, we use the well-known set of multicommodity network flow CANAD instances (Frangioni, 2012). Like the authors of Hewitt *et al.* (2013), we use the set identified as "C" to study the unsplittable MCND problem. This set consists of 31 instances with varying fixed-to-variable cost ratios and a mix of tightly- and loosely-capacitated arcs. The instances range in size with 20-30 nodes, 230-700 arcs, and 40-400 commodities.

Table 1 – *Definitions of the valid inequality configurations reported. Configurations (b), (c), and (d) add the listed inequalities to (a).*

Conf	Definition
(a)	LP of (1)
(b)	Linking constraints
(c)	k-split c-strong inequalities for $k \leq 10$ (Atamtürk & Rajan, 2002)
(d)	SAC-Pack constraints (5)

To examine how these cuts perform in comparison to existing methods, we compare the strengthened LP relaxation solutions after adding disaggregated capacity linking constraints, k-split c-strong inequalities with k set to a maximum value of 10, and our new valid inequalities (5). We use the nomenclature as defined in Table 1. In Table 2, we provide a summary of the results obtained from solving the CANAD instances. Specifically, for each configuration tested, we provide the average values across all 31 instances for the best objective, LP relaxation solution (LPR), number of cuts added, cut generation time in seconds, IP gap, and the improvement of the IP gap. We also include the number of instances where the optimal objective was achieved after adding the cuts (# Opt). To calculate the IP gap, we use:

$$\frac{(\text{best obj} - \text{LPR})}{\text{best obj}} \tag{6}$$

with the best objective set to the best found IP objective after solving the model for 2 hours using Gurobi default settings. Thus, the resulting average best objective used was $194,527$ with an average IP gap of 0.96%.

Table 2 – *Summarized results for the CANAD instances.*

Conf	LPR	Cuts	Cut Time (s)	IP Gap	Impr	# Opt
(a)	163,189	0	0	16.1%	0.0%	0
(b)	185,747	93,245	18	4.5%	72.0%	0
(c)	188,162	1,705	161	3.3%	79.7%	2
(d)	191,107	3,467	125	1.8%	89.1%	2
(d),(c)	191,507	4,091	246	1.6%	90.4%	3

The results in Table 2 confirm that our new valid inequalities can significantly strengthen arc-based formulations. Our first observation is that the disaggregated capacity linking constraints alone (Conf (b)) led to an improvement of over 70% for the base LP, but this improvement comes at the cost of adding an average of over 90,000 extra constraints. We next observe that while the k-split c-strong inequalities (c) are able to achieve an improvement of 80% with just a fraction of the cuts compared to (b), the SAC-Pack constraints (d) can achieve an even greater improvement of almost 90%, reducing the IP gap of (c) by almost half. We also note that while roughly double the number of k-split c-strong inequalities are generated, the total time to generate the SAC-Pack constraints is still less than that required to generate the k-split c-strong constraints, on average. We next tested the configuration where k-split c-strong inequalities were generated after adding all SAC-Pack constraints, leading to a marginally improved IP gap. However, in general, it seems that the SAC-Pack constraints (d) achieve the right balance of cuts added and time required for the improvements gained.

We additionally compared configurations (d) and (d),(c) to the final bound found in Hewitt *et al.* (2013) (reported in their Table 8). This final bound was found after dynamically adding disaggregated capacity linking constraints and cover inequalities to their extended LP formulation (through user cuts) at the root node and then executing their branch-and-price procedure for 30 minutes. We found that 21 and 29 instances (out of 31) achieved better bounds for configurations (d) and (d),(c), respectively, when compared to the final bounds found in Hewitt *et al.* (2013). While this is a bit of an unfair comparison (due to the advances in modern day commercial solvers, which help in the separation routines of SAC-Pack inequalities, compared to those available at the time Hewitt *et al.* (2013) was written), we report this finding purely to demonstrate the potential improvements our new valid inequalities could lead to in such branch-and-price procedures.

References

Atamtürk, A., & Rajan, D. 2002. On splittable and unsplittable flow capacitated network design arc–set polyhedra. *Mathematical Programming*, **92**, 315–333.

Bienstock, D., Chopra, S., Günlük, O., & Tsai, C.Y. 1998. Minimum cost capacity installation for multicommodity network flows. *Mathematical programming*, **81**, 177–199.

Chen, L., Chen, W.K., Yang, M.M., & Dai, Y.H. 2021. An exact separation algorithm for unsplittable flow capacitated network design arc-set polyhedron. *Journal of Global Optimization*, **81**, 659–689.

Crainic, T.G., & Gendreau, M. 2021. Heuristics and metaheuristics for fixed-charge network design. *Chap. 14, pages 427–467 of:* Crainic, T. G., Gendreau, M., & Gendron, B. (eds), *Network Design with Applications to Transportation and Logistics*. Springer.

Frangioni, A. 2012. *Multicommodity Flow Problems*. April 2024.

Hewitt, M., Nemhauser, G., & Savelsbergh, MWP. 2013. Branch-and-price guided search for integer programs with an application to the multicommodity fixed-charge network flow problem. *INFORMS Journal on Computing*, **25**(2), 302–316.