

Article

A New Bounding Procedure for Transportation Problems with Stepwise Costs

Jingyi Liu

Department of Mathematics, College of Sciences, Northeastern University, Shenyang 110819, China;
liujingyi@mail.neu.edu.cn

Abstract

Transportation planning often involves not only shipment costs but also setup costs associated with deploying vehicles or transport resources. In many practical logistics operations, this setup cost does not remain constant but increases stepwise with the number of vehicles used, reflecting economies of scale and scheduling thresholds. To capture this realistic feature, this study investigates the transportation problem with stepwise costs, where total costs combine shipment-dependent variable costs and vehicle activation costs. We develop a mixed-integer programming (MIP) model to represent the problem and propose an efficient algorithm based on variable-splitting reformulation and a row-and-column generation scheme. This approach dynamically introduces only the necessary variables and constraints, enabling the solution of large-scale instances that are otherwise computationally challenging. Numerical experiments show that the method produces high-quality solutions much faster than direct MIP solvers. The results highlight the model's practical value in optimizing fleet utilization and transportation cost structures in real logistics and supply chain systems.

Keywords: transportation problem; stepwise cost; discretized reformulation; row-and-column generation; variable neighborhood descent

MSC: 90B06; 90C11; 90C10; 65K05



Academic Editor: Rafael Guillermo García-Cáceres

Received: 3 October 2025

Revised: 8 November 2025

Accepted: 14 November 2025

Published: 19 November 2025

Citation: Liu, J. A New Bounding Procedure for Transportation Problems with Stepwise Costs. *Mathematics* **2025**, *13*, 3709. <https://doi.org/10.3390/math13223709>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the era of globalization, enterprises are increasingly exposed to complex and volatile market environments. Under such conditions, competition among firms is not only reflected in product quality or pricing but also in the efficiency and sophistication of supply chain management. Within this framework, logistics supply chains are often regarded as the lifeline of enterprises, as they encompass the entire material flow from raw material procurement to final product delivery. Among the various functions of logistics, transportation plays a critical role, acting as the essential link that connects upstream and downstream processes. Effective transportation services enable the smooth coordination of supply and demand, improve responsiveness to market fluctuations, and ultimately enhance overall competitiveness.

Given its central role in determining both operational efficiency and overall profitability, the fixed-charge transportation problem (FCTP) has been one of the most widely studied models in operations research and logistics management [1]. Over several decades, it has attracted extensive scholarly attention and produced a rich body of theoretical results,

as well as practical insights that continue to shape transportation planning and decision-making in industry. In its classical formulation, the FCTP assumes that transportation costs consist of two components: a variable cost that grows linearly with the amount shipped, and a setup cost that is triggered as soon as a shipping route is used, regardless of the load transported. This structure offers a convenient abstraction, enabling elegant linear programming models and efficient solution methods.

In practice, however, real-world logistics systems involve important additional complexities that the classical model does not capture. Most notably, shipments are executed by vehicles with finite capacity, and each vehicle that is deployed incurs a setup or activation cost that does not depend on how much of its capacity is actually utilized. Consequently, the cost for moving goods between a supply location and a demand location increases in a stepwise fashion, as the number of required vehicles rises with the total shipment volume. The cost function, combining a linear variable component with a discontinuous stepwise fixed component, captures an important feature abstracted from real-world logistics problems and motivates our theoretical investigation. Beyond this, actual logistics systems are more complex, involving dynamic demand, multimodal transportation, and other intricate factors that are also worth studying.

This stepwise cost structure makes the problem fundamentally different from the classical FCTP. While the FCTP already introduces combinatorial complexity through binary route-activation decisions, the vehicle-based cost structure adds another layer of difficulty, as the model must determine not only whether a route is used but also how many capacity-constrained vehicles are required. These characteristics significantly increase the modeling complexity and hinder the scalability of exact algorithms. However, this practically relevant setting has received limited attention, leaving a research gap between theoretical transportation models and real-world logistics systems where setup costs vary in discrete steps. As FCTP is a well-known NP-hard problem [2], and TPSC constitutes a strict generalization of FCTP, the NP-hardness of TPSC follows immediately. Consequently, solution times grow rapidly with problem size, motivating the development of new modeling and optimization techniques to bridge theory and practice in transportation planning.

To address these challenges, this paper makes the following contributions:

- We formulate a mixed-integer programming (MIP) model. Structural properties of optimal solutions are analyzed, revealing that they may be non-basic and providing insights for neighborhood design in local search.
- To overcome the weak relaxation of the base model, we propose a variable discretization reformulation and two classes of valid inequalities that significantly tighten bounds. A row-and-column generation (RCG) algorithm is then developed to dynamically introduce only essential variables and constraints for large-scale instances.
- A variable neighborhood descent (VND) is proposed to obtain the feasible solution from the fractional solution via rounding and refine it through problem-specific neighborhoods.
- Extensive experiments on benchmark instances demonstrate that the proposed method markedly improves lower bounds, yields near-optimal feasible solutions, and consistently outperforms direct MIP solvers in both quality and efficiency.

The remainder of this paper is organized as follows. Section 2 reviews related work on fixed-charge transportation problems. Section 3 introduces the problem definition and formulates the baseline mixed-integer programming model, together with the structural properties of optimal solutions. Section 4 presents the reformulation, valid inequalities, and the proposed bounding framework, which combines a row-and-column generation algorithm with a VND-based local search. Section 5 reports the computational results, and Section 6 concludes the paper with remarks and directions for future research.

2. Related Literature

The transportation problem has long been recognized as a fundamental research topic with broad applications in social and economic activities. Since the pioneering formulation by Hitchcock [3] in the context of production organization and rail transport, a substantial body of work has emerged. Subsequent contributions by Kantorovitch [4] and Chanas et al. [5] further enriched both the theoretical foundation and practical applications of TP. As research progressed and practical needs evolved, numerous variants of TP were introduced. In this study, we focus on the fixed-charge transportation problem (FCTP) and review the relevant literature in this domain.

Hirsch and Dantzig [6] first introduced the FCTP, establishing its mathematical formulation and proving that optimal solutions lie at extreme points of the feasible region. Building on this foundation, researchers have proposed numerous extensions and solution approaches. For example, Yang and Feng [7] considered a time-constrained FCTP that incorporates both cost and time dimensions, solved via tabu search. Buson et al. [8] developed a reduced-cost iterated local search heuristic for large-scale instances, while Fisk and McKeown [9] examined the pure FCTP and proposed a direct search method that exploits the embedded 0–1 knapsack structure. Mondal et al. [10] formulated multi-objective FCTP models with quantity-dependent costs and discount policies, solved via hybrid metaheuristics based on the artificial bee colony algorithm. Kartli [11] proposed integrated heuristic–metaheuristic algorithms that enhance solution quality for medium-sized instances. Zhu et al. [12] established new NP-hardness results and approximation schemes for the pure FCTP, whereas Allahdadi and Rivaz [13] extended FCTP formulations to rough interval uncertainty, broadening linear programming applications under imprecise data.

More complex logistics environments have also been explored. Adlakha et al. [14] designed a more-for-less heuristic for large-scale assignment-type FCTPs. Klose [15] studied the single-source variant of the problem. Xie and Jia [16] introduced a nonlinear FCTP, which was solved using a hybrid genetic algorithm. Safi and Razmjoo [17] considered interval-valued parameters and proposed solution schemes based on ordering relations of interval numbers. Pramanik et al. [18] modeled two-stage supply chain networks under Gaussian fuzzy settings, while Giri et al. [19] investigated fully fuzzy multi-item FCTPs. Paraskevopoulos et al. [20] studied multi-commodity transportation networks with fixed charges and proposed hybrid heuristics for effective solutions. Balaji et al. [21] introduced capacity constraints, solving them with genetic algorithms and simulated annealing. Cosma et al. [22,23] studied two-stage supply chain and transportation problems with fixed costs, introducing hybrid genetic algorithms and parallel methods that improve solution quality and scalability. From an exact optimization perspective, Legault et al. [24] developed a novel reformulation for the single-sink FCTP, significantly tightening relaxations and improving branch-and-cut efficiency. Most recently, Kartli et al. [25] proposed a new heuristic for the classical FCTP, demonstrating superior performance on benchmark tests.

Most of these studies assume that fixed costs are triggered solely by activating a route, independent of shipment size. In practice, however, transportation often involves more complex structures. For example, when inventory constraints or vehicle capacities require multi-stage shipments, fixed costs may increase once a shipment exceeds certain volume thresholds. This gives rise to the so-called step or two-step FCTP. Kowalski and Lev [26] analyzed structural properties of this variant, while El-Sherbiny [27] formulated it mathematically and proposed a mutation-based artificial immune algorithm for its solution. Collectively, these works illustrate complementary progress through metaheuristics, parallelization, and reformulation-based exact methods, offering valuable insights for tackling complex stepwise fixed-charge transportation problems. However, most existing approaches remain limited to simplified stepwise or two-step settings and do not explicitly

capture the discrete activation of capacity-constrained vehicles. Moreover, few studies combine lower-bound strengthening with high-quality feasible solution generation in a unified framework, leaving room for methodological improvement.

This study differs from prior work by extending the two-step FCTP to a more realistic setting that explicitly incorporates vehicle capacity limits and activation-based stepwise costs. In the proposed formulation, each deployed vehicle incurs a fixed cost regardless of load, making the total fixed transportation cost a stepwise function of the number of vehicles used. This structure better reflects real logistics operations and introduces new modeling and computational challenges that existing approaches have not addressed.

3. Problem Formulation

To rigorously analyze the TPSC, we now present its formal problem formulation, including problem description, mixed-integer programming model, structural properties of optimal solutions, and its linear relaxation.

3.1. Problem Description

The TPSC can be formally described as follows. Consider a set of n supply nodes and a set of m demand nodes. The production capacity at each supply node and the demand requirement at each demand node are known in advance. Goods produced at any supply node can be delivered to any demand node. A single type of transport vehicle is used, each with limited capacity Q . During transportation, vehicles are loaded as fully as possible; however, when the volume transported is less than the vehicle capacity, one vehicle must still be deployed. The transportation cost consists of two parts: (i) a variable cost, which is proportional to the shipment volume, and (ii) a setup cost, which corresponds to the activation of a vehicle. Each time a vehicle is used, a setup cost is incurred regardless of whether the vehicle is fully loaded. The objective is to minimize the total transportation cost so that all supply, demand, and capacity constraints are satisfied.

Figure 1a shows the cost structure of the classical FCTP, where transport costs consist of variable costs proportional to shipment volume and fixed costs that are independent of volume but incurred whenever a route is used. The two-step FCTP is an extension of this model, which has been comprehensively studied by Kowalski et al. [26]. Its cost structure is illustrated in Figure 1b, where parameter A denotes a threshold of transported volume. If the shipment does not exceed A , the fixed cost is k_1 ; if it exceeds A , an additional fixed cost k_2 is incurred. Figure 1c depicts the cost function of the TPSC. As shown, the total setup cost depends on the number of vehicles activated: each deployed vehicle generates a fixed cost, and even if the vehicle is not fully loaded, the full setup cost of one vehicle is still applied.

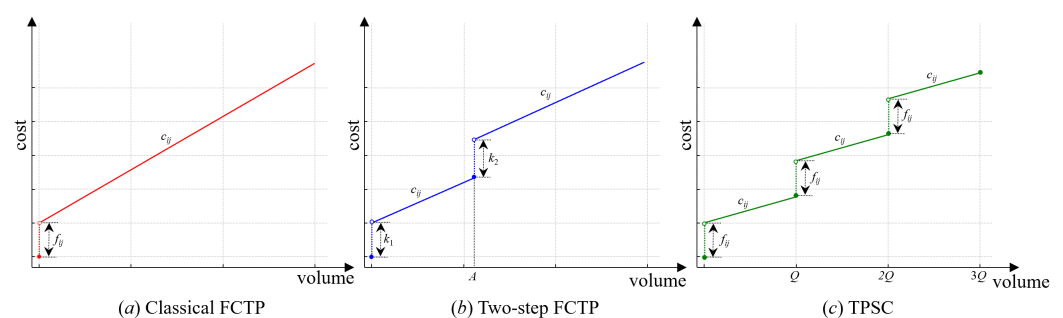


Figure 1. Cost structures of three different transportation problems.

3.2. Mixed-Integer Programming Model

The symbols and notation appearing in the model and in later sections are listed as follows:

- S : set of supply nodes, indexed by i .
- T : set of demand nodes, indexed by j .
- E : set of arcs from supply to demand, indexed by e .
- n : number of supply nodes.
- m : number of demand nodes.
- c_{ij} : unit transportation cost from supply node i to demand node j .
- f_{ij} : fixed setup cost of deploying one vehicle from supply node i to demand node j .
- a_i : available supply at node i .
- b_j : demand requirement at node j .
- v_{ij} : maximum transportable quantity from supply node i to demand node j , defined as $v_{ij} = \min\{a_i, b_j\}$.
- Q : loading capacity of a single vehicle.
- x_{ij} : quantity shipped from supply node i to demand node j .
- y_{ij} : number of vehicles used to transport goods from i to j .

We assume that all supply quantities a_i , demand requirements b_j , and vehicle capacity Q are integer-valued. Moreover, the total supply is required to match the total demand, i.e., $\sum_{i \in S} a_i = \sum_{j \in T} b_j$, which ensures the feasibility of the flow balance constraints. With these modeling assumptions and notation, a MIP formulation of the TPSC is given as follows:

$$(\mathbf{F}_0) \quad \min \quad \sum_{i \in S} \sum_{j \in T} (c_{ij} x_{ij} + f_{ij} y_{ij}) \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in T} x_{ij} = a_i, \forall i \in S \quad (2)$$

$$\sum_{i \in S} x_{ij} = b_j, \forall j \in T \quad (3)$$

$$x_{ij} \leq Q y_{ij}, \forall i \in S, \forall j \in T \quad (4)$$

$$0 \leq x_{ij} \leq v_{ij}, \forall i \in S, \forall j \in T \quad (5)$$

$$y_{ij} \in \mathbb{Z}_+, \forall i \in S, \forall j \in T \quad (6)$$

The objective function (1) minimizes the total transportation cost, consisting of variable costs proportional to shipment volumes and fixed setup costs incurred by vehicle activation. Constraint (2) enforces supply conservation by requiring that the total outbound flow from each supply node equals its available production. Constraint (3) ensures demand satisfaction at each demand node. Constraint (4) imposes the vehicle capacity limit by linking the shipped volume on each route to the number of activated vehicles. Constraint (5) defines the feasible domain of shipment volumes. Finally, constraint (6) specifies that the number of vehicles is a nonnegative integer.

3.3. Properties of the Optimal Solution

Before presenting the reformulation and algorithmic framework, we first investigate the structural properties of the optimal solution, which provide important insights for model strengthening and algorithm design.

Proposition 1. *For the TPSC, an optimal solution is not necessarily a basic feasible solution (BFS) of the associated linear programming relaxation.*

Proof. To verify this property, consider the following instance. $|S| = |T| = 2$, $a_1 = 20$, $a_2 = 10$, $b_1 = 10$, $b_2 = 20$, $Q = 3$, $c_{11} = 10$, $c_{12} = 10$, $c_{21} = 20$, $c_{22} = 10$, $f_{11} = f_{12} = f_{21} = f_{22} = 30$. The balance constraints (2) and (3) yield $x_{12} = 20 - x_{11}$, $x_{21} = 10 - x_{11}$, $x_{22} = x_{11}$, $0 \leq x_{11} \leq 10$. Hence, the feasible region of the transportation polytope is the line segment $\{(x_{11}, 20 - x_{11}, 10 - x_{11}, x_{11}) : 0 \leq x_{11} \leq 10\}$, whose extreme points (and thus basic feasible solutions) occur at the endpoints: $x^{B_1} = (10, 10, 0, 10)$ and $x^{B_2} = (0, 20, 10, 0)$. Therefore, this instance admits exactly two BFS.

Let $x' = \lambda x^{B_1} + (1 - \lambda)x^{B_2} = (10\lambda, 20 - 10\lambda, 10(1 - \lambda), 10\lambda)$ with $0 < \lambda < 1$, so that x' is a strict convex combination of x^{B_1} and x^{B_2} , and hence a feasible solution. Its objective value is

$$z(x') = (400 - 100\lambda) + 30 \cdot \left(\left\lceil \frac{10\lambda}{3} \right\rceil + \left\lceil \frac{20-10\lambda}{3} \right\rceil + \left\lceil \frac{10(1-\lambda)}{3} \right\rceil + \left\lceil \frac{10\lambda}{3} \right\rceil \right) \quad (7)$$

Since the numerators of the terms inside the ceiling functions (10λ , $20 - 10\lambda$, and $10(1 - \lambda)$) share a common multiple of 10, the ceiling values change only when these numerators cross multiples of 3. Consequently, the ceiling terms vary only at the breakpoints $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. On each open interval between breakpoints, the second term of (7) is constant, and the first term strictly decreases in λ ; therefore, the minimum over $(0, 1)$ is reached at a breakpoint. Evaluating $z(\lambda)$ at these breakpoints gives $z(0.1) = 750$, $z(0.2) = 710$, $z(0.3) = 700$, $z(0.4) = 720$, $z(0.5) = 680$, $z(0.6) = 670$, $z(0.7) = 690$, $z(0.8) = 650$, $z(0.9) = 640$. Hence, the minimum over $0 < \lambda < 1$ is achieved at $\lambda = 0.9$, with $x' = (9, 11, 1, 9)$ and $z(x') = 640$.

The objective values of BFSs x^{B_1} and x^{B_2} are as follows:

$$z(x^{B_1}) = 10 \cdot (10 + 10 + 10) + 30 \cdot \left(\left\lceil \frac{10}{3} \right\rceil + \left\lceil \frac{10}{3} \right\rceil + \left\lceil \frac{10}{3} \right\rceil \right) = 660, \quad (8)$$

$$z(x^{B_2}) = 10 \cdot 20 + 20 \cdot 10 + 30 \cdot \left(\left\lceil \frac{20}{3} \right\rceil + \left\lceil \frac{10}{3} \right\rceil \right) = 730. \quad (9)$$

Thus, we have

$$z(x') = 640 < \min\{660, 730\} = \min\{z(x^{B_1}), z(x^{B_2})\}. \quad (10)$$

Since x' has four positive components while $m + n - 1 = 3$, it is not a BFS. Therefore, there exists an optimal solution that is not a basic feasible solution. $\square$

The above example demonstrates that, unlike the classical transportation problem (TP), where every optimal LP solution can be chosen as a basic feasible solution (BFS), the transportation problem with stepwise costs (TPSC) may admit an optimal solution that is feasible but not basic. This structural difference carries two important implications. First, simplex-type algorithms, which traverse from one BFS to another, may fail to encounter certain optimal points that arise as convex combinations of BFSs, revealing a richer and more intricate relaxation geometry. Second, the existence of non-BFS optimal solutions provides theoretical motivation for designing advanced algorithms that move beyond simplex pivots. Since fractional vehicle variables y_{ij} naturally appear in the relaxation, this structure can be exploited to construct tailored valid inequalities, rounding rules, or neighborhood moves within metaheuristic frameworks. In particular, the fact that optimality may occur at non-extreme points highlights the value of polyhedral strengthening and decomposition-based methods over pure simplex or direct branching approaches.

3.4. Linear Relaxation Model

In order to obtain a valid lower bound for F_0 , we consider its LP relaxation, denoted by LF_0 . The LP relaxation is obtained by relaxing the integrality requirements of the vehicle variables y_{ij} while keeping all other constraints unchanged. This relaxation provides a tractable approximation of the original problem and serves as the basis for further reformulation and strengthening techniques developed in the subsequent sections.

From constraint (4), $x_{ij} \leq Q y_{ij}$, and from constraint (5), $x_{ij} \leq v_{ij}$. The shipment on arc (i, j) must satisfy both: if $Q \leq v_{ij}$, the capacity $Q y_{ij}$ dominates; otherwise the bound v_{ij} is binding. The linear relaxation of y_{ij} thus depends on the relation between v_{ij} and Q .

Figure 2 illustrates how the relaxed transportation cost varies with shipment volume under the two cases. When $v_{ij} < Q$, the cost curve in Figure 2a corresponds to the red line, representing the objective of the linear relaxation. Conversely, when $v_{ij} \geq Q$, the curve in Figure 2b corresponds to the lower blue line, which depicts the relaxed objective function, ensuring that it never exceeds the original piecewise linear cost function for any shipment volume.

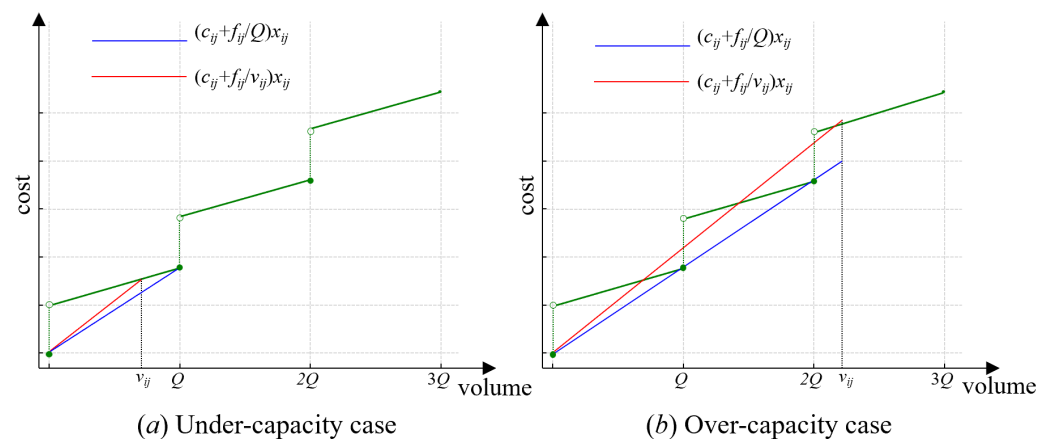


Figure 2. The relaxed transportation cost changes as the shipment volume increases.

We define an auxiliary variable $\bar{y}_{ij}$ to represent the relaxed number of vehicles under the two cases:

$$\bar{y}_{ij} = \begin{cases} \frac{x_{ij}}{Q}, & \text{if } v_{ij} \geq Q, \\ \frac{x_{ij}}{v_{ij}}, & \text{if } v_{ij} < Q. \end{cases} \quad (11)$$

In summary, the linear relaxation model (LF_0) can be written as

$$(LF_0) \quad \min \sum_{i \in S} \sum_{j \in T} \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})} \right) x_{ij} \quad (12)$$

$$\text{s.t.} \quad \sum_{j \in T} x_{ij} = a_i, \forall i \in S \quad (13)$$

$$\sum_{i \in S} x_{ij} = b_j, \forall j \in T \quad (14)$$

In LF_0 , the coefficients in constraints (13) and (14) are all equal to one, so the constraint matrix is totally unimodular. Since both the supply values a_i and the demand values b_j are integers, the relaxation LF_0 always admits an integer optimal solution.

4. Solution Method

To tackle the computational intractability of the mixed-integer model F_0 on large-scale instances—and the weak bounds provided by its linear relaxation LF_0 —this section develops a reformulation-based bounding procedure. We first apply a variable splitting scheme to obtain an equivalent discretized model F_1 , which preserves optimality while exposing additional structure. We then strengthen the linear relaxation LF_1 by introducing two families of valid inequalities—namely rounding inequalities and subset-cover inequalities—to tighten the lower bound and shrink the feasible region of the relaxation. Building on the reconstructed model, we design an RCG algorithm that starts from a restricted master problem and dynamically adds only the necessary rows and columns. This procedure iteratively produces improved lower bounds and dual solutions, enabling scalable exact optimization for large problem instances.

4.1. Discretized Reformulation

By discretizing the shipment volume on each route, the continuous decision variable x_{ij} in the model F_0 is represented by a family of binary variables, which simplifies the model structure and exposes additional combinatorial properties. After incorporating valid inequalities, the discretized model can be efficiently solved using the RCG algorithm to obtain strong lower bounds as well as dual information to guide the search direction.

Based on model F_0 , the transportation network is discretized into a set of arcs representing possible shipment quantities between supply and demand nodes. For any arc (i, j) , the feasible shipment volume q lies in $[1, v_{ij}]$, where $v_{ij} = \min\{a_i, b_j\}$; when $q = 0$, no goods are transported along that route. For each feasible shipment volume q , we define a binary variable:

$$\eta_{ij}^q = \begin{cases} 1, & \text{if the transported quantity from } i \text{ to } j \text{ equals } q, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the shipment variable x_{ij} in F_0 can be expressed as

$$x_{ij} = \sum_{q=1}^{v_{ij}} q \eta_{ij}^q. \quad (15)$$

Let g_{ij}^q denote the total cost (variable plus fixed) when q units are shipped from i to j , i.e.,

$$g_{ij}^q = \begin{cases} c_{ij}q + f_{ij} \left\lceil \frac{q}{Q} \right\rceil, & \text{if } v_{ij} \geq Q \\ c_{ij}q + f_{ij}, & \text{if } v_{ij} < Q. \end{cases} \quad (16)$$

The discretized model F_1 is then formulated as

$$(\mathbf{F}_1) \quad \min \quad \sum_{i \in S} \sum_{j \in T} \sum_{q=1}^{v_{ij}} g_{ij}^q \eta_{ij}^q \quad (17)$$

$$\text{s.t.} \quad \sum_{j \in T} \sum_{q=1}^{v_{ij}} q \eta_{ij}^q = a_i, \forall i \in S, \quad (18)$$

$$\sum_{i \in S} \sum_{q=1}^{v_{ij}} q \eta_{ij}^q = b_j, \forall j \in T, \quad (19)$$

$$\eta_{ij}^q \in \{0, 1\}, \forall (i, j) \in E, q = 1, \dots, v_{ij}. \quad (20)$$

The objective function (17) minimizes the total transportation cost across all supply–demand pairs. Constraints (18) ensure that the total quantity shipped from each supply node i equals its available supply a_i , while constraints (19) guarantee that each demand node j receives the required quantity b_j . Finally, constraints (20) define binary variables η_{ij}^q , indicating whether the q -th replicated arc of (i, j) is used.

Relaxing the constraint (20) yields the linear relaxation, denoted as LF_1 . We next present several propositions that describe its key properties.

Proposition 2. *The linear relaxations LF_0 and LF_1 have the same optimal value, i.e., $z(LF_0) = z(LF_1)$.*

Proof. Let $\hat{\eta}$ be an optimal solution to LF_1 . In LF_1 , the binary constraints on η_{ij}^q are relaxed to $0 \leq \eta_{ij}^q \leq 1$. Define the induced shipment variables $\hat{x}_{ij} = \sum_{q=1}^{v_{ij}} q \hat{\eta}_{ij}^q$, $(i, j) \in E$. Since $\hat{\eta}$ satisfies (18) and (19), it follows that $\hat{x}$ satisfies the supply and demand balance constraints of LF_0 , (13) and (14). Hence, the induced vector $\hat{x}$ is feasible to LF_0 .

For each $(i, j) \in E$ and each $q \leq v_{ij}$, using (16) we obtain

$$g_{ij}^q = \begin{cases} c_{ij}q + f_{ij}\lceil q/Q \rceil \geq \left(c_{ij} + \frac{f_{ij}}{Q}\right)q, & v_{ij} \geq Q, \\ c_{ij}q + f_{ij} \geq \left(c_{ij} + \frac{f_{ij}}{v_{ij}}\right)q, & v_{ij} < Q, \end{cases} \quad (21)$$

which yields

$$\sum_{q=1}^{v_{ij}} g_{ij}^q \hat{\eta}_{ij}^q \geq \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})}\right) \sum_{q=1}^{v_{ij}} q \hat{\eta}_{ij}^q = \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})}\right) \hat{x}_{ij}. \quad (22)$$

Summing over all arcs $(i, j) \in E$ gives

$$\sum_{(i,j) \in E} \sum_{q=1}^{v_{ij}} g_{ij}^q \hat{\eta}_{ij}^q \geq \sum_{(i,j) \in E} \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})}\right) \hat{x}_{ij}. \quad (23)$$

Since $\hat{\eta}$ is optimal for LF_1 , the left-hand side is exactly $z(LF_1)$. The right-hand side equals the objective value of LF_0 evaluated at its feasible solution $\hat{x}$. Therefore, the optimal value of LF_1 is no smaller than that of LF_0 , i.e., $z(LF_1) \geq z(LF_0)$.

We now prove the converse. Let $\hat{x}$ be any feasible solution to LF_0 . We construct $\hat{\eta}$ such that $\sum_{q=1}^{v_{ij}} q \hat{\eta}_{ij}^q = \hat{x}_{ij}$ for all $(i, j) \in E$.

For arcs with $v_{ij} < Q$, set $\hat{\eta}_{ij}^{v_{ij}} = \hat{x}_{ij}/v_{ij}$ and $\hat{\eta}_{ij}^q = 0$ for $q \neq v_{ij}$. Then $0 \leq \hat{\eta}_{ij}^{v_{ij}} \leq 1$, $\sum_q q \hat{\eta}_{ij}^q = \hat{x}_{ij}$, and

$$\sum_q g_{ij}^q \hat{\eta}_{ij}^q = (c_{ij}v_{ij} + f_{ij}) \frac{\hat{x}_{ij}}{v_{ij}} = \left(c_{ij} + \frac{f_{ij}}{v_{ij}}\right) \hat{x}_{ij}. \quad (24)$$

For arcs with $v_{ij} \geq Q$, let $m = \lfloor v_{ij}/Q \rfloor$ and $r = \hat{x}_{ij}/Q$. Define $t_k \in [0, 1]$ for $k = m, m-1, \dots, 1$ by

$$t_k = \min\left\{1, \frac{r - \sum_{\ell=k+1}^m \ell t_\ell}{k}\right\}, \quad (25)$$

so that $\sum_{k=1}^m k t_k = r$. Set $\hat{\eta}_{ij}^{kQ} = t_k$ for $k = 1, \dots, m$, and $\hat{\eta}_{ij}^q = 0$ for all other q . Then $\sum_q q \hat{\eta}_{ij}^q = \hat{x}_{ij}$, and using $g_{ij}^{kQ} = c_{ij}(kQ) + f_{ij}k$,

$$\sum_q g_{ij}^q \hat{\eta}_{ij}^q = \sum_{k=1}^m (c_{ij}kQ + f_{ij}k) t_k = \left(c_{ij} + \frac{f_{ij}}{Q}\right) \hat{x}_{ij}. \quad (26)$$

Thus, in both cases,

$$\sum_q g_{ij}^q \hat{\eta}_{ij}^q = \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})}\right) \hat{x}_{ij}, \quad \forall (i, j) \in E. \quad (27)$$

Summing over all arcs,

$$\sum_{(i,j) \in E} \sum_q g_{ij}^q \hat{\eta}_{ij}^q = \sum_{(i,j) \in E} \left(c_{ij} + \frac{f_{ij}}{\min(Q, v_{ij})}\right) \hat{x}_{ij}. \quad (28)$$

The left-hand side is the objective value of LF_1 at $\hat{\eta}$, and the right-hand side is the objective value of LF_0 at $\hat{x}$. Since $\hat{x}$ was arbitrary among feasible solutions of LF_0 , we conclude that $z(LF_1) \leq z(LF_0)$.

Combining the two inequalities proves $z(LF_1) = z(LF_0)$. $\square$

The lower bound provided by LF_1 coincides with that of LF_0 . Although LF_1 introduces additional binary-structured variables and is therefore less attractive to solve directly for bounding purposes, it enables the introduction of strong valid inequalities. This structural advantage makes LF_1 a more effective foundation for tightening relaxations and improving the quality of lower bounds.

4.2. Valid Inequalities

The Chvátal–Gomory (CG) cut [28] is a classical cutting-plane technique for integer programming. Given a valid linear inequality $u^\top x \leq \beta$ for the LP relaxation and an integer vector x , the inequality $\lfloor u \rfloor^\top x \leq \lfloor \beta \rfloor$ is also valid for all integer-feasible x .

Based on constraint (18), for any supply site $i \in S$, define a positive integer parameter σ within the range $1 < \sigma < a_i$. According to the CG-cut procedure, by applying floor rounding on a term-by-term basis, we obtain the following valid inequality:

$$\sum_{j \in T} \sum_{q=1}^{v_{ij}} \left\lfloor \frac{q}{\sigma} \right\rfloor \eta_{ij}^q \leq \left\lfloor \frac{a_i}{\sigma} \right\rfloor, \quad \forall (i, \sigma) \in \mathcal{R}^1, \quad (29)$$

where $\mathcal{R}^1$ denotes the set of all pairs (i, σ) , i.e., $\mathcal{R}^1 = \{(i, \sigma) : i \in S, \sigma \in \mathbb{Z}_+, 1 < \sigma < a_i\}$.

By applying the same procedure to constraint (19), we obtain another valid inequality:

$$\sum_{i \in S} \sum_{q=1}^{v_{ij}} \left\lfloor \frac{q}{\sigma} \right\rfloor \eta_{ij}^q \leq \left\lfloor \frac{b_j}{\sigma} \right\rfloor, \quad \forall (j, \sigma) \in \mathcal{R}^2, \quad (30)$$

where $\mathcal{R}^2$ denotes the set of all pairs (j, σ) , i.e., $\mathcal{R}^2 = \{(j, \sigma) : j \in T, \sigma \in \mathbb{Z}_+, 1 < \sigma < b_j\}$.

Moreover, for inequalities (29) and (30), replacing the floor operator with the ceiling operator reverses the inequality direction, thereby yielding two additional valid inequalities:

$$\sum_{j \in T} \sum_{q=1}^{v_{ij}} \left\lceil \frac{q}{\sigma} \right\rceil \eta_{ij}^q \geq \left\lceil \frac{a_i}{\sigma} \right\rceil, \quad \forall (i, \sigma) \in \mathcal{R}^1, \quad (31)$$

$$\sum_{i \in S} \sum_{q=1}^{v_{ij}} \left\lceil \frac{q}{\sigma} \right\rceil \eta_{ij}^q \geq \left\lceil \frac{b_j}{\sigma} \right\rceil, \quad \forall (j, \sigma) \in \mathcal{R}^2. \quad (32)$$

Subset-cover inequalities, originally studied in the context of set-covering [29] and vehicle-routing problems [30], constitute a well-known family of strong valid inequalities for capacitated problems. They exploit the relationship between limited supply (or capacity) and excessive demand to identify infeasible fractional solutions that satisfy the basic balance constraints but cannot correspond to any feasible integer solution. By enforcing that the excess demand of any subset of customers must be covered by vehicles or supplies outside a chosen subset, these inequalities significantly tighten the LP relaxation and are widely used in branch-and-cut algorithms for capacitated network design and vehicle routing.

The core idea of the subset-cover inequality in our context is as follows. Let $K \subseteq S$ denote a subset of supply nodes and $L \subseteq T$ denote a subset of demand nodes. If the total demand of L exceeds the total supply of K , i.e.,

$$\sum_{j \in L} b_j > \sum_{i \in K} a_i, \quad (33)$$

then part of the demand in L must necessarily be satisfied by supplies from $S \setminus K$. Consequently, the number of vehicles dispatched from $S \setminus K$ to L cannot be less than the shortage between the two quantities divided by the vehicle capacity Q .

Formally, let $\mathcal{C}$ denote the set of all pairs (K, L) satisfying condition (33). For every $(K, L) \in \mathcal{C}$, the following subset-cover inequality holds:

$$\sum_{i \in S \setminus K} \sum_{j \in L} y_{ij} \geq \left\lceil \frac{\sum_{j \in L} b_j - \sum_{i \in K} a_i}{Q} \right\rceil, \quad \forall (K, L) \in \mathcal{C}. \quad (34)$$

By noting that $y_{ij} = \sum_{q=1}^{v_{ij}} \left\lceil \frac{q}{Q} \right\rceil \eta_{ij}^q$, inequality (34) can be equivalently reformulated as

$$\sum_{i \in S \setminus K} \sum_{j \in L} \sum_{q=1}^{v_{ij}} \left\lceil \frac{q}{Q} \right\rceil \eta_{ij}^q \geq \left\lceil \frac{\sum_{j \in L} b_j - \sum_{i \in K} a_i}{Q} \right\rceil, \quad \forall (K, L) \in \mathcal{C}. \quad (35)$$

By incorporating inequalities (29)–(32) and (35) into the constraints of $\mathbf{LF}_1$, we obtain the relaxed model with valid inequalities, denoted as $\mathbf{LF}_2$.

4.3. Lower Bounding Procedure

In the $\mathbf{LF}_2$ model, there are $\sum_{i=1}^n \sum_{j=1}^m v_{ij}$ variables and at least $|m| + |n| + 2 \sum_{i=1}^n (a_i - 2) + 2 \sum_{j=1}^m (b_j - 2) + |\mathcal{C}|$ constraints. Even for small-scale instances, the number of constraints becomes extremely large. In particular, there exist exponentially many combinations (K, L) , implying that $|\mathcal{C}|$ grows exponentially. Therefore, although $\mathbf{LF}_2$ is a linear program, solving it directly to optimality (or even near-optimality) in a reasonable time is highly challenging for standard optimization solvers such as CPLEX. To address this difficulty, we solve it within an RCG framework.

At each iteration, only a restricted subset of columns (variables) and rows (valid inequalities) of $\mathbf{LF}_2$ are maintained in a restricted master problem (RMP). The RMP, which can be regarded as a tractable approximation of the full linear relaxation, is solved to optimality, yielding both primal and dual solutions. Based on the dual information, new variables with negative reduced costs are identified by evaluating the reduced costs.

Let α_i and β_j denote the dual variables associated with the supply and demand balance constraints (18) and (19); $\phi_{i\sigma}$ and $\psi_{j\sigma}$ the dual variables corresponding to the CG-cuts (29) and (30) with “ $\leq$ ” inequalities; $\xi_{i\sigma}$ and $\zeta_{j\sigma}$ the dual variables associated with the CG-cuts

(31) and (32) with “ $\geq$ ” inequalities; and γ_{KL} the dual variables corresponding to the subset-cover inequalities (35). The reduced cost of each variable η_{ij}^q excluded from the current RMP is computed as follows.

$$RC_{ij}^q = g_{ij}^q - q\alpha_i - q\beta_j - \sum_{\sigma:(i,\sigma) \in \bar{\mathcal{R}}^1} \left\lfloor \frac{q}{\sigma} \right\rfloor \phi_{i\sigma} - \sum_{\sigma:(j,\sigma) \in \bar{\mathcal{R}}^2} \left\lfloor \frac{q}{\sigma} \right\rfloor \psi_{j\sigma} - \sum_{\sigma:(i,\sigma) \in \bar{\mathcal{R}}^1} \left\lceil \frac{q}{\sigma} \right\rceil \xi_{i\sigma} \\ - \sum_{\sigma:(j,\sigma) \in \bar{\mathcal{R}}^2} \left\lceil \frac{q}{\sigma} \right\rceil \zeta_{j\sigma} - \sum_{(K,L) \in \bar{\mathcal{C}}: i \notin K, j \in L} \left\lceil \frac{q}{Q} \right\rceil \gamma_{KL}, \forall (i, j, q) \in \mathcal{V} \setminus \bar{\mathcal{V}}. \quad (36)$$

Here, $\bar{\mathcal{R}}^1$ denotes the index set of the cuts of types (29) and (31) that have already been generated, $\bar{\mathcal{R}}^2$ denotes the index set of the already generated cuts of types (30) and (32), and $\bar{\mathcal{C}}$ denotes the index set of the already generated subset-cover cuts (35). $\mathcal{V}$ and $\bar{\mathcal{V}}$ denote the index sets corresponding to all variables and to the variables that have already been generated, respectively. Columns with negative reduced cost, identified through (36), are added to the RMP. Once no further columns with negative reduced cost can be found, the procedure switches to row generation.

For each of the four families of CG cuts (29)–(32), the most violated inequality is identified by examining the sets $\mathcal{R}^1 \setminus \bar{\mathcal{R}}^1$ and $\mathcal{R}^2 \setminus \bar{\mathcal{R}}^2$. Since there exist exponentially many combinations (K, L) , identifying the most violated subset-cover inequality (35) is not straightforward. To achieve this, a dedicated separation problem must be solved.

Given an optimal solution $\bar{\eta}$ to the RMP, the combination $(K^*, L^*) \in \mathcal{C} \setminus \bar{\mathcal{C}}$ that most strongly violates inequality (34) can be identified by solving the following auxiliary MIP formulation. Here, $\bar{\theta}_{ij} = \sum_q \bar{\eta}_{ij}^q$ represents the total assignment flow from supply i to demand j in the solution $\bar{\eta}$; u_i is a binary variable for each supply node i , with $u_i = 0$ if and only if $i \in K^*$ and $u_i = 1$ otherwise; v_j is a binary variable for each demand node j , with $v_j = 1$ if and only if $j \in L^*$ and $v_j = 0$ otherwise; and δ_{ij} is a binary variable equal to 1 if and only if $i \notin K^*$ and $j \in L^*$, and 0 otherwise. The model is formulated as follows.

$$(\mathbf{SP}) \quad z^* = \min \sum_{i \in S} \sum_{j \in T} \bar{\theta}_{ij} \delta_{ij}, \quad (37)$$

$$\text{s.t.} \quad \delta_{ij} \geq u_i + v_j - 1, \quad \forall (i, j) \in E, \quad (38)$$

$$\sum_{i \in S} a_i u_i + \sum_{j \in T} b_j v_j \geq \sum_{i \in S} a_i + 1, \quad (39)$$

$$\delta_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E, \quad (40)$$

$$u_i \in \{0, 1\}, \quad \forall i \in S, \quad (41)$$

$$v_j \in \{0, 1\}, \quad \forall j \in T. \quad (42)$$

By solving **SP** via a MIP solver, if the optimal objective value satisfies $z^* < 1$, then for the optimal solution η of **LF**₁, there exists a subset pair $(K^*, L^*) \in \mathcal{C}$ for which inequality (34) is valid, where $K^* = \{i \in S : u_i = 0\}$ and $L^* = \{j \in T : v_j = 1\}$. Otherwise, inequality (34) is not applicable. For further details, the reader is kindly referred to [2], which provides a comprehensive treatment of related techniques for separating subset-cover inequalities in the context of the FCTP.

The iterative process of generating columns and rows continues until no column with a negative reduced cost and no violated inequality can be identified. At this stage, the algorithm converges and terminates, yielding a valid lower bound for the original problem. Notably, the integration of dynamic variable generation with adaptive cut management enables the restricted master problem to remain computationally tractable while progressively approaching the strength of the full relaxation. The overall RCG procedure, which formalizes this process and ensures convergence, is summarized in Algorithm 1.

Algorithm 1 Row-and-Column Generation for LF_2

```

1: Initialization: Solve  $LF_0$  to obtain an integer solution  $X$  and construct initial columns.
2: repeat
3:   Solve the restricted master problem (RMP).
4:   Compute reduced costs of non-active variables.
5:   if negative reduced cost variables exist then
6:     Add corresponding columns to RMP.
7:   end if
8:   Check for violated valid inequalities from the cut families (29)–(32).
9:   if violated inequalities exist then
10:    Add them to RMP.
11:  end if
12:  Solve the subproblem (SP) to separate violated subset-cover inequalities.
13:  if violated inequalities exist then
14:    Add them to RMP.
15:  end if
16: until no negative reduced cost and no violated inequality are found
17: Output: Optimal solution of  $LF_2$  and a valid lower bound.

```

Beyond its algorithmic role, the proposed RCG algorithm also provides managerial insights into the structural characteristics of the problem. By selectively generating variables and constraints, the procedure naturally highlights the dominant transportation patterns and the most critical capacity or cover relations. This not only improves computational efficiency but also offers a deeper understanding of which resources and routes primarily drive the overall solution. From a practical perspective, such information can guide decision makers in identifying bottlenecks, prioritizing capacity expansions, or redesigning allocation rules. Moreover, the sparsity of the final active set of columns and cuts suggests that only a small subset of decisions is truly essential for optimality, which can be exploited to design faster heuristics and more interpretable decision-support tools. In this sense, the RCG procedure goes beyond providing tight bounds—it also reveals actionable structural insights that enhance both planning efficiency and managerial decision-making.

4.4. Upper Bounding Procedure

The upper bound is obtained through a rounding procedure that transforms the fractional solution of the RMP into a feasible primal solution. This procedure guarantees feasibility with respect to supply and demand constraints and yields a valid, albeit not necessarily tight, upper bound for the problem. However, such solutions are seldom optimal due to the discrete stepwise cost structure and the fixed-charge characteristics of the model. To further improve solution quality, we introduce a local search (LS) procedure that iteratively explores neighborhoods specifically tailored to the structure of (F_0) . By applying cycle adjustments, opening or closing vehicles, and rebalancing flows, the LS procedure systematically reduces total cost while maintaining feasibility. The connection between RCG and VND procedures is shown in Figure 3.

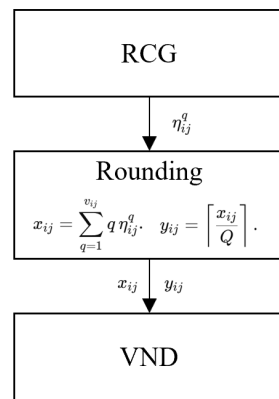


Figure 3. The connection between RCG and VND procedures.

Let $X = (x, y)$ denote a feasible solution to (F_0) . A neighborhood move transforms X into another feasible solution $X' = (x', y')$. Four neighborhoods are considered:

- N1 (Cycle Adjustment in Open Arcs). Let $C = \{(i_1, j_1), (i_2, j_1), \dots, (i_k, j_k), (i_1, j_k)\}$ be an even-length alternating cycle in the bipartite graph of open arcs, partitioned into “positive” C^+ and “negative” C^- . For $\theta > 0$:

$$x'_{ij} = \begin{cases} x_{ij} + \theta, & (i, j) \in C^+, \\ x_{ij} - \theta, & (i, j) \in C^-, \\ x_{ij}, & \text{otherwise,} \end{cases} \quad (43)$$

subject to $0 \leq x'_{ij} \leq v_{ij}$ and $x'_{ij} \leq Qy_{ij}$. The cost variation is

$$\Delta = \theta \left(\sum_{(i,j) \in C^+} c_{ij} - \sum_{(i,j) \in C^-} c_{ij} \right). \quad (44)$$

- N2 (Vehicle Opening with Cycle Adjustment). Consider a saturated or closed arc (i, j) . Opening a new vehicle increases $y_{ij} \leftarrow y_{ij} + 1$, enlarging capacity by Q and incurring fixed cost f_{ij} . Inserting (i, j) into an alternating cycle C , the cost change is

$$\Delta = f_{ij} + \theta \left(\sum_{(p,q) \in C^+} c_{pq} - \sum_{(p,q) \in C^-} c_{pq} \right). \quad (45)$$

- N3 (Vehicle Closing). For an open arc (i, j) with $y_{ij} > 0$, if the load of one vehicle can be rerouted through feasible cycles, then $y'_{ij} = y_{ij} - 1$. The cost change is

$$\Delta = -f_{ij} + \Delta^{\text{reroute}}, \quad (46)$$

where Δ^{reroute} denotes the net change in linear transportation costs.

- N4 (Intra-Arc Rebalancing). Within an arc (i, j) , flow can be redistributed across vehicles. If this eliminates one vehicle, then $y'_{ij} = y_{ij} - 1$, and the net cost variation is

$$\Delta = -f_{ij} + \Delta^{\text{linear}}, \quad (47)$$

where Δ^{linear} is the change in linear costs due to reallocation.

We adopt a VND framework that explores the neighborhoods in increasing order of complexity. The overall upper bounding procedure is summarized in Algorithm 2

Algorithm 2 Upper Bounding Procedure for F_0

- 1: **Step 1 (Initialization):** Start from a feasible solution (x, y) obtained by the rounding procedure.
- 2: **Step 2 (Local Search):**
- 3: **repeat**
- 4: Construct the neighborhood of (x, y) using:
 - **Cycle Adjustment (N1):** Select an alternating cycle C in the bipartite network of open arcs. Adjust flows along C with step size θ , preserving feasibility.
 - **Vehicle Opening (N2):** For a saturated or closed arc (i, j) , increase $y_{ij} \leftarrow y_{ij} + 1$ (open a new vehicle of capacity Q), insert (i, j) into a cycle C , and adjust flows accordingly.
 - **Vehicle Closing (N3):** For an open arc (i, j) with $y_{ij} > 0$, attempt to reroute the load of one vehicle through cycles. If successful, set $y_{ij} \leftarrow y_{ij} - 1$.
 - **Intra-Arc Rebalancing (N4):** Within a single arc (i, j) , redistribute flow among vehicles. If one vehicle becomes redundant, update $y_{ij} \leftarrow y_{ij} - 1$.
- 5: Evaluate all candidate moves and select the best improving neighbor (x', y') .
- 6: **if** (x', y') improves upon (x, y) **then**
- 7: Update $(x, y) \leftarrow (x', y')$.
- 8: **end if**
- 9: **until** no improving neighbor is found or stopping criteria are met
- 10: **Output:** Improved feasible solution (x, y) and its objective value.

To improve computational efficiency, several acceleration strategies are incorporated into the local search framework. These strategies include candidate-list restriction, first-improvement and incremental evaluation, adaptive neighborhood ordering, and occasional diversification with parallel evaluation to avoid premature convergence, detailed as follows:

- **Candidate list reduction:** restrict neighborhood exploration to the K most promising arcs, selected by indicators such as the marginal transportation cost c_{ij} or the effective cost $c_{ij} + f_{ij}/Q$, thereby focusing computational effort on moves with the greatest potential benefit.
- **First-improvement rule:** accept the first improving move encountered, which reduces the number of evaluations per iteration and accelerates convergence compared with a best-improvement strategy.
- **Incremental evaluation:** maintain auxiliary data structures such as residual capacities $r_{ij} = Qy_{ij} - x_{ij}$ and arc-specific cost contributions, enabling incremental evaluation of candidate moves in $O(|C|)$ time.
- **Adaptive neighborhood ordering:** adopt a VND scheme that restarts from N1 whenever an improvement is found, ensuring that simple and computationally cheap adjustments are exploited before invoking more complex neighborhoods.
- **Diversification:** apply periodic perturbations or controlled restarts to enlarge the explored solution space, thereby avoiding premature convergence to poor-quality local optima.

5. Numerical Results

In this section, we present computational experiments to evaluate the proposed framework that integrates the RCG-based lower-bounding procedure with the VND-based upper-bounding procedure. The objectives are threefold: (i) to examine the impact of valid inequalities on strengthening the LP relaxation and to analyze the scalability and efficiency of the RCG algorithm, (ii) to evaluate the ability of VND to enhance solution quality by improving upper bounds, and (iii) to provide a comprehensive performance comparison between the proposed framework and a state-of-the-art commercial MIP solver in terms

of solution quality, computational time, and scalability. The benchmark instances and computational environment are described below.

A suite of benchmark instances was generated following the procedure of Glover et al. [31], with modifications to incorporate the stepwise cost characteristics of the TPSC. The numbers of supply and demand nodes were fixed at $n = 50$ and $m = 100$. Supply quantities were uniformly distributed within $[400, 800]$, while demands were sampled from $[200, 400]$ under the condition of overall balance. Variable transportation costs were drawn from $[4, 8]$. The vehicle capacity was set at two levels, $Q \in \{50, 100\}$, and the setup cost for deploying a vehicle was sampled from four intervals: $[100, 400]$, $[200, 800]$, $[400, 1600]$, and $[800, 3200]$. Combining the two capacity levels with the four setup-cost intervals yields eight instance classes, denoted as A–H. Each class contains 15 independent instances, resulting in a total of 120 test instances that provide representative and diverse scenarios for performance evaluation.

All algorithms were implemented in C++ and compiled with Visual Studio 2023 (64-bit). LP and MIP models were solved using IBM ILOG CPLEX 20.1.0 with default parameter settings. All tests were executed on a Windows 10 workstation equipped with an Intel Xeon Platinum 2.4 GHz CPU, 16 GB RAM, and a 10 MB cache.

5.1. Effectiveness of Valid Inequalities and RCG Performance

This subsection examines the effect of valid inequalities on strengthening the LP relaxation. We compare two models: (i) the baseline formulation ($\mathbf{LF}_0$) and (ii) the discretized reformulation augmented with both classes of valid inequalities ($\mathbf{LF}_2$), solved by the RCG procedure. The evaluation focuses on three aspects: the tightness of the lower bounds, the relative improvement over the baseline, and the additional computational effort incurred.

Table 1 reports the average results over 15 instances for each class (A–H). For $\mathbf{LF}_0$, it lists the number of variables (Vars), the number of constraints (Constr.), the optimal objective value providing a valid lower bound (LB) for the original problem, and the computation time (CPU(s)). For $\mathbf{LF}_2$, it reports the number of columns (Cols) and cuts (Cuts) generated by the RCG procedure, together with the corresponding lower bound (LB) and computation time (CPU(s)); note that Cols and Cuts refer only to those generated in the restricted master problem, rather than the full set of variables and constraints in the complete formulation. The last column presents the relative improvement in the lower bound (Δ_{LB}) obtained by $\mathbf{LF}_2$ over $\mathbf{LF}_0$.

Table 1. Comparison of lower bounds and computation times between $\mathbf{LF}_0$ and $\mathbf{LF}_2$.

Instances	$\mathbf{LF}_0$				$\mathbf{LF}_2$				Δ_{LB} (%)
	Vars	Constr.	LB	CPU(s)	Cols	Cuts.	LB	CPU(s)	
A	5000	150	372,112	0.01	852	2338	398,526	107	7.10
B	5000	150	414,680	0.01	931	2889	442,685	244	6.75
C	5000	150	788,565	0.01	1592	3064	854,425	173	8.35
D	5000	150	1,315,305	0.01	1767	3585	1,408,339	323	7.07
E	5000	150	242,430	0.01	1940	3271	266,388	179	9.88
F	5000	150	289,470	0.01	1054	3556	308,372	204	6.53
G	5000	150	480,300	0.01	1216	2632	515,102	141	7.25
H	5000	150	742,440	0.01	1016	3488	804,379	273	8.34

As reported in Table 1, the baseline formulation $\mathbf{F}_0$, when directly relaxed, provides relatively weak lower bounds that severely underestimate the problem's true difficulty. By contrast, enhancing the discretized formulation with valid inequalities ($\mathbf{LF}_2$) substantially strengthens the relaxation, yielding relative improvements in the lower bound between

6.53% and 9.88% across all instance classes. This tightening effect highlights the crucial role of the inequalities in bridging the gap left by F_0 's weak relaxation. The improvement, however, comes at the expense of increased computation time, which grows from negligible levels in LF_0 to several hundred seconds in LF_2 . Within the row-and-column generation (RCG) framework, the inequalities are introduced dynamically as additional cuts, while new allocation variables are generated as columns. This iterative enrichment yields a much tighter relaxation with far fewer variables and constraints than the full model, thereby maintaining scalability. By avoiding explicit enumeration of the entire discretized problem, RCG ensures that the enhanced formulation remains tractable for larger instances, confirming its role as a critical building block of the proposed framework.

Beyond these numerical gains, the results also offer methodological insights. First, the proposed inequalities effectively exploit problem structure: rounding inequalities eliminate fractional vehicle activations, while subset-cover inequalities rule out infeasible partial assignments. Second, the improvements remain stable across all instance classes, indicating that the inequalities generalize well across problem scales. Finally, by providing stronger lower bounds, the inequalities not only tighten the relaxation but also enhance the efficiency of the RCG process, as improved bounds accelerate pruning and focus the search more effectively.

5.2. Upper-Bound Improvement via VND

This subsection evaluates the quality of feasible solutions obtained by the proposed upper-bounding procedure. Two strategies are considered: (a) direct rounding of the fractional RMP solution, and (b) an enhanced approach that augments rounding with a VND-based local search exploiting problem-specific neighborhoods. The evaluation focuses on three performance indicators: the resulting upper bound (UB), the duality gap relative to the best lower bound obtained by solving LF_2 with RCG, and the computation time (CPU(s)). Table 2 reports these measures for each instance class under both methods, with the last column Δ_{UB} summarizing the relative improvement in solution quality achieved by VND.

Table 2. Comparison of upper bounds obtained by rounding with VND-based local search.

Instances	Rounding			VND			Δ_{UB}
	UB	Gap (%)	CPU(s)	UB	Gap (%)	CPU(s)	
A	405,497	1.75	0.01	402,924	1.10	121	0.63
B	464,092	4.84	0.01	458,324	3.53	107	1.24
C	873,825	2.27	0.01	869,031	1.71	177	0.55
D	1,515,824	7.63	0.01	1,480,451	5.12	189	2.33
E	279,504	4.92	0.01	270,328	1.48	160	3.28
F	321,285	4.19	0.01	313,277	1.59	142	2.49
G	539,020	4.64	0.01	530,137	2.92	186	1.65
H	853,243	6.07	0.01	836,369	3.98	149	1.98

As shown in Table 2, the rounding-only approach generates feasible solutions almost instantaneously, but the quality of these solutions is limited, with duality gaps between 6.85% and 9.67%. By incorporating VND-based local search, the upper bounds are consistently improved by several percentage points, leading to significantly smaller gaps, all below 4%. Importantly, this improvement is achieved with only modest additional runtime, which remains within three seconds even for the largest tested instances. These results demonstrate that the integration of local search into the upper-bounding procedure provides a clear and consistent benefit without sacrificing computational efficiency.

The findings also underscore the complementary nature of the two strategies. Rounding ensures rapid feasibility with negligible cost, serving as a reliable starting point, while local search leverages problem-specific neighborhoods—such as cycle adjustments and vehicle rebalancing—to systematically refine the solution. This combination not only delivers high-quality upper bounds but also narrows the duality gap to a practical level, reinforcing the scalability and robustness of the proposed framework.

5.3. Comparison with a State-of-the-Art MIP Solver

Finally, we present a comprehensive comparison between the proposed RCG&VND framework and a state-of-the-art commercial MIP solver applied directly to F_0 . The evaluation considers three aspects: (i) the quality of lower and upper bounds, (ii) the duality gap of final solutions, and (iii) the computational runtime across all instance classes.

Table 3 reports, for both CPLEX and RCG&VND, the lower bound (LB), the upper bound (UB), the optimality gap (Gap), and the computation time (CPU(s)). In addition, two performance indicators are included: $\Delta_{UB}(\%) = 100 \times (UB_{CPLEX} - UB_{VND}) / UB_{CPLEX}$, where a positive value indicates that VND achieves a better (lower) UB than CPLEX, and $\Delta_{LB}(\%) = 100 \times (LB_{RCG} - LB_{CPLEX}) / LB_{CPLEX}$, where a positive value indicates that RCG achieves a better (higher) LB than CPLEX. The optimality gap is calculated as $100 \times (UB - LB) / UB$.

Table 3. Comprehensive Comparison between RCG&VND and CPLEX.

Instances	CPLEX				RCG&VND				$\Delta_{UB}(\%)$	$\Delta_{LB}(\%)$
	LB	UB	Gap (%)	CPU(s)	LB	UB	Gap (%)	CPU(s)		
A	394,295	408,523	3.61	3600	398,526	402,924	1.10	228	1.37	1.07
B	439,313	462,855	5.36	3600	442,685	458,324	3.53	351	0.98	0.77
C	837,316	872,838	4.24	3600	854,425	869,031	1.71	350	0.44	2.04
D	1,396,903	1,499,496	7.34	3600	1,408,339	1,480,451	5.12	512	1.27	0.82
E	262,843	278,491	5.95	3600	266,388	270,328	1.48	339	2.93	1.35
F	304,757	319,050	4.69	3600	308,372	313,277	1.59	346	1.81	1.19
G	511,536	537,761	5.13	3600	515,102	530,137	2.92	327	1.42	0.70
H	796,474	851,746	6.94	3600	804,379	836,369	3.98	422	1.81	0.99

Table 3 provides a comprehensive comparison between the proposed RCG&VND framework and the commercial solver CPLEX. Under the 1-h time limit, CPLEX fails to close the duality gaps, which remain between 3.6% and 7.3% across the tested instances. In contrast, RCG&VND consistently yields tighter bounds: the lower bounds are improved by up to 2.0%, and the upper bounds are reduced by up to 2.9%, leading to significantly smaller gaps of 1.1–3.9%. These improvements stem from the fact that the proposed inequalities tighten the LP relaxation, raising the lower bound early, while the RCG procedure adds only the most influential variables and constraints, keeping the relaxation small and efficient. The VND-based local search then focuses on correcting the key structural sources of suboptimality, improving the upper bound. This coordinated tightening of both bounds enables convergence within about 600 s, compared with the full 3600-s budget required by CPLEX.

These findings demonstrate the effectiveness of combining lower-bound strengthening, row-and-column generation, and problem-specific local search. The framework not only narrows the duality gap but also produces high-quality feasible solutions within practical computation times. Overall, RCG&VND achieves a favorable balance between bound

quality, solution accuracy, and scalability, clearly outperforming a state-of-the-art general-purpose solver on large-scale instances of the stepwise cost transportation problem.

6. Conclusions

This paper examined the transportation problem with stepwise costs, arising when vehicle setup costs increase with the number of vehicles deployed. We formulated a mixed-integer programming model, analyzed its structural properties, and addressed the weakness of its linear relaxation through a variable-splitting reformulation and valid inequalities. To handle the large scale of the discretized model, we designed an RCG algorithm that dynamically introduces only the most relevant variables and constraints. In addition, feasible solutions were generated via a rounding heuristic and subsequently improved through VND-based local search. Computational experiments on benchmark instances demonstrated that the proposed framework produces substantially tighter bounds, achieves high-quality feasible solutions, and consistently outperforms direct MIP solvers in both solution quality and efficiency. The study provides theoretical insights into stepwise cost modeling and a practical framework for large-scale transportation planning with vehicle activation costs.

Despite its theoretical value, the proposed model has certain limitations in practical applicability. Future research may build upon this work in several directions. Methodologically, richer neighborhoods and adaptive hybrid heuristics could be integrated into the local search procedure to improve convergence, while alternative polyhedral relaxations and decomposition schemes may yield tighter bounds. From an application perspective, extending the framework to address stochastic demand, multi-modal routing, or integrated production–distribution settings would further enhance its applicability and strengthen its relevance to complex supply chain planning problems with discrete and nonlinear cost structures.

Funding: This work was supported by Liaoning Provincial Science and Technology Plan—Joint Program of Key Research and Development Project (2023JH2/101800046).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Calvete, H.I.; Galé, C.; Iranzo, J.A.; Toth, P. A Matheuristic for the Two-Stage Fixed-Charge Transportation Problem. *Comput. Oper. Res.* **2018**, *95*, 113–122. [\[CrossRef\]](#)
2. Roberti, R.; Bartolini, E.; Mingozzi, A. The fixed charge transportation problem: An exact algorithm based on a new integer programming formulation. *Manag. Sci.* **2015**, *61*, 1275–1291. [\[CrossRef\]](#)
3. Hitchcock, F.L. The distribution of a product from several sources to numerous localities. *Stud. Appl. Math.* **1941**, *20*, 224–230. [\[CrossRef\]](#)
4. Kantorovitch, L. On the translocation of masses. *Manag. Sci.* **1958**, *5*, 1–4. [\[CrossRef\]](#)
5. Chanas, S.; Kołodziejczyk, W.; Machaj, A. A fuzzy approach to the transportation problem. *Fuzzy Sets Syst.* **1984**, *13*, 211–221. [\[CrossRef\]](#)
6. Hirsch, W.M.; Dantzig, G.B. The fixed charge problem. *Nav. Res. Logist.* **1968**, *15*, 413–424. [\[CrossRef\]](#)
7. Yang, L.; Feng, Y. A bicriteria solid transportation problem with fixed charge under stochastic environment. *Appl. Math. Model.* **2007**, *31*, 2668–2683. [\[CrossRef\]](#)
8. Buson, E.; Roberti, R.; Toth, P. A reduced-cost iterated local search heuristic for the fixed-charge transportation problem. *Oper. Res.* **2014**, *62*, 1095–1106. [\[CrossRef\]](#)

9. Fisk, J.; McKeown, P. The pure fixed charge transportation problem. *Nav. Res. Logist.* **2015**, *26*, 631–641. [[CrossRef](#)]
10. Mondal, M.; Mandal, G.; Bhunia, A.K. Investigation of a multi-objective fixed charge transportation problem with quantity dependent transportation cost and discount policy via metaheuristics. *INFOR* **2025**, *63*, 562–591. [[CrossRef](#)]
11. Kartli, N. Hybrid algorithms for fixed charge transportation problem. *Kybernetika* **2025**, *61*, 141–167. [[CrossRef](#)]
12. Zhu, P.F.; Chen, G.T.; Zhang, A. On the pure fixed charge transportation problem. *Discrete Optim.* **2025**, *55*, 100–115. [[CrossRef](#)]
13. Allahdadi, M.; Rivaz, S. Some new results on rough interval linear programming problems and their application to scheduling and fixed-charge transportation problems. *RAIRO—Oper. Res.* **2024**, *58*, 3697–3714. [[CrossRef](#)]
14. Adlakha, V.; Kowalski, K.; Vemuganti, R.R.; Lev, B. More-for-less algorithm for fixed-charge transportation problems. *Omega* **2007**, *35*, 116–127. [[CrossRef](#)]
15. Klose, A. Algorithms for solving the single-sink fixed-charge transportation problem. *Comput. Oper. Res.* **2008**, *35*, 2079–2092. [[CrossRef](#)]
16. Xie, F.; Jia, R. Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm. *Comput. Ind. Eng.* **2012**, *63*, 763–778. [[CrossRef](#)]
17. Safi, M.R.; Razmjoo, A. Solving fixed charge transportation problem with interval parameters. *Appl. Math. Model.* **2013**, *37*, 8341–8347. [[CrossRef](#)]
18. Pramanik, S.; Jana, D.K.; Mondal, S.K.; Maiti, M. A fixed-charge transportation problem in two-stage supply chain network in Gaussian type-2 fuzzy environments. *Inf. Sci.* **2015**, *325*, 190–214. [[CrossRef](#)]
19. Giri, P.K.; Maiti, M.K.; Maiti, M. Fully fuzzy fixed charge multi-item solid transportation problem. *Appl. Soft Comput.* **2015**, *27*, 77–91. [[CrossRef](#)]
20. Paraskevopoulos, D.C.; Bektaş, T.; Crainic, T.G.; Potts, C.N. A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. *Eur. J. Oper. Res.* **2016**, *253*, 265–279. [[CrossRef](#)]
21. Balaji, A.N.; Nilakantan, J.M.; Nielsen, I.; Jawahar, N.; Ponnambalam, S.G. Solving fixed charge transportation problem with truck load constraint using metaheuristics. *Ann. Oper. Res.* **2017**, *273*, 207–236. [[CrossRef](#)]
22. Cosma, O.; Pop, P.C.; Sabo, C. An Efficient Hybrid Genetic Approach for Solving the Two-Stage Supply Chain Network Design Problem with Fixed Costs. *Mathematics* **2020**, *8*, 712. [[CrossRef](#)]
23. Cosma, O.; Pop, P.C.; Danciulescu, D. A Parallel Algorithm for Solving a Two-Stage Fixed-Charge Transportation Problem. *Informatica* **2020**, *31*, 681–706. [[CrossRef](#)]
24. Legault, R.; Cote, J.F.; Gendron, B. A novel reformulation for the single-sink fixed-charge transportation problem. *Math. Program.* **2023**, *202*, 169–198. [[CrossRef](#)]
25. Kartli, N.; Bostanci, E.; Guzel, M.S. A New Algorithm for Optimal Solution of Fixed Charge Transportation Problem. *Kybernetika* **2023**, *59*, 45–63. [[CrossRef](#)]
26. Kowalski, K.; Lev, B. On step fixed-charge transportation problem. *Omega* **2008**, *36*, 913–917. [[CrossRef](#)]
27. El-Sherbiny, M.M. Alternate mutation based artificial immune algorithm for step fixed charge transportation problem. *Egypt. Inform. J.* **2012**, *13*, 123–134. [[CrossRef](#)]
28. Chvátal, V. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math.* **1973**, *4*, 305–337. [[CrossRef](#)]
29. Balas, E.; Zemel, E. Facets of the knapsack polytope from minimal covers. *SIAM J. Appl. Math.* **1978**, *34*, 119–148. [[CrossRef](#)]
30. Jepsen, M.; Petersen, B.; Spoorendonk, S.; Pisinger, D. Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows. *Oper. Res.* **2008**, *56*, 497–511. [[CrossRef](#)]
31. Glover, F.; Amini, M.; Kochenberger, G. Parametric ghost image processes for fixed-charge problems: A study of transportation networks. *J. Heuristics* **2005**, *11*, 307–336. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.