공학박사학위논문

Single Path Multicommodity Trading Problem on Acyclic Network: Polyhedral Structure and Approximability

무회로 네트워크에서의 단일경로 다품종거래문제의 해법 연구: 다면체의 구조와 근사가능성을 중심으로

2023 년 8 월

서울대학교 대학원

산업공학과

임 세 호

Single Path Multicommodity Trading Problem on Acyclic Network: Polyhedral Structure and Approximability

무회로 네트워크에서의 단일경로 다품종거래문제의 해법 연구: 다면체의 구조와 근사가능성을 중심으로

지도교수 홍 성 필

이 논문을 공학박사 학위논문으로 제출함

2023 년 7 월

서울대학교 대학원

산업공학과

임 세 호

임세호의 공학박사 학위논문을 인준함

2023 년 7 월

위 원 장	홍 유 석	(인)
부위원장	홍 성 필	(인)
위 원	이 경 식	(인)
위 원	류 홍 서	(인)
위 원	안 형 찬	(인)

Abstract

Single Path Multicommodity Trading Problem on Acyclic Network: Polyhedral Structure and Approximability

Seho Yim
Department of Industrial Engineering
The Graduate School
Seoul National University

The Single Path Multicommodity Trading Problem on Acyclic Network(sMTP) involves finding a single path on an acyclic network that maximizes profits by selectively transporting goods between pairs of nodes as long as the carrying volume does not exceed the capacity. The operator can earn a profit for fulfilling each transportation request, but they also incur costs that are proportional to the volume of transportation for every unit of distance traveled. The objective is to select a path that passes through some of the nodes, maximizes profit, and subtracts the logistic cost from the revenue. In this paper, we explore the structure of the polyhedron and investigate the approximability of sMTP. Our Study involves both theoretical analysis based on integer programming and the development of approximation algorithms.

Theoretical analysis is conducted by first presenting two models for sMTP, along with several families of valid inequalities for each model. We also identify the condi-

tions under which these inequalities serve as facet-defining inequalities and propose efficient separation algorithms. Next, we delve into the approximability of the problem. We discuss the inherent limitations in developing approximation algorithms and address both the inapproximability and the integrality gap. To pave the way for an approximation algorithm, we focus on specific cases and devise approximation algorithms tailored to those scenarios. Building upon these algorithms, we present an approximation algorithm for sMTP. Furthermore, we explore the applicability of the techniques utilized in our proposed approximation algorithms to other related problems. Lastly, we conduct a comparative analysis to evaluate the performance of the algorithms derived from our findings.

Keywords: Freight Logistics, Integer Programming, Valid Inequality, Inapproximability, Approximation Algorithm

Student Number: 2017-20407

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we consider the Single Path Multicommodity Trading Problem on Acyclic Network (sMTP). In the situation where a vehicle operator travels from their origin to their destination in an acyclic network, they want to make a profit by making transportation between different regions en route. It is possible to perform only some of the transportation requests between regions. Each request earns revenue according to the unit weight transported. On the other hand, logistic costs proportional to the cargo load occur per unit distance movement of the vehicle. sMTP is a problem of determining which regions to visit and which transportation to perform between each region in order to maximize profit, defined as revenue minus cost.

Problem 1.0.1. Single Path Multicommodity Trading Problem on Acyclic Network(sMTP) sMTP is a sextuple (G, c, D, d, p, U) where $G = (V, A)$ is an directed acyclic graph with $V = \{1, \cdots n\}$ and $A = \{ij \mid i < j,\ i, j \in V\}$. There is a non-negative logistic cost c_{ij} incurred by a unit flow on arc $ij \in A$. There is a set $D \subseteq V \times V$ of *s-d pairs* (k, l) with $k < l$ such that there is a nonobligatory demand $d(k, l)$ on a product from node k to node l. Vehicle operator gets a revenue $r(k, l)$ from a unit weight of product traded between s-d pair (k, l). Also denote by

U the capacity, namely the maximum weight of products that can be carried by the vehicle. The problem is to find a path $P = i_1 - i_2 - \cdots - i_p$ and the trading volumes $x_{i_u i_v}$, $1 \leq u < v \leq p$ between s-d pairs along P that maximizes the profit, i.e. the total revenue minus the total logistic cost.

sMTP arises in various situations. First, a situation similar to the problem itself can occur. For example, in Dong et al. (2022), a similar problem was considered to efficiently use cargo in a backhaul situation where the cargo reaches the existing destination and returns to the original location. In a situation where the owner of the goods to be transported and the owner of the vehicle are the same, it is common to solve the problem of transporting all goods to the necessary locations using the vehicle. However, recently, with the emergence of food delivery platforms, outsourcing of transportation has become more common, and the situation where each vehicle owner pursues profit and accepts only some of the requests has increased. If the utility of transportation for the transportation requester is as much as the revenue, ultimately, each vehicle plays a role in increasing the total utility, and therefore, this can be seen as a problem of maximizing marginal utility. From this perspective, it can be seen that sMTP also appears in the situation of a public transportation system.

The public transportation system is an attractive option for many people who need to travel from their origin to their destination, either in terms of time or cost. Individuals choose different modes of transportation such as train, metro, bus, or airplane based on their financial and time constraints, origin-destination pairs, and the available transportation infrastructure. To establish a public transportation system, different infrastructures must be installed depending on the means, such as stations,

railroads, or vehicles, which incurs significant fixed costs. Due to these high costs, infrastructure related to public transportation tends to be expanded in multiple horizons over time. However, planning multiple horizons at once is extremely difficult, and there are already various facilities established nowadays. Alternatively, when planning to introduce a new public transport we consider what sets it apart from the services provided by existing transports. If it allows for faster travel between two points that were relatively time-consuming in the existing transportation system, the effect can be significant. On the other hand, the larger the volume of people for whom the service is effective, the greater the effect. Let us consider a situation of introducing new express train. For instance, if there were only regular trains that stopped at every station along a railroad path, introducing an express train that stops only at stations with high traffic can make travel between those stations faster than regular trains. However, it should be noted that the more stations an express train stops at, the more beneficiaries of the service there will be, but the longer the travel time within the same section will be due to more stops. When considering the trade-off between these two factors, it becomes clear that choosing which stations to stop at is an import decision. An train network $G = (V, A)$ consists of n stations $V = \{1, \cdots, n\}$ and a railroad $A = \{ij | 1 \leq i < j \leq n\}$ that allows movement between the stations. For each pair of origin-destination stations, or OD pairs, $(k, l)(k, l \in V)$, there exists travel demand $d(k, l)$. The travel time from station k to station l in the current system, only with regular trains, is denoted by $r(k, l)$. Now, we aim to operate a new express train system in which the train departs from station 1 and arrives at station n, passing through certain stations of the route. If the train passes through station j after station i, the travel time between them is denoted

by c_{ij}. Thus, if the express train passes through $v_0(=1) - v_1 - \cdots - v_k(=n)$, the passengers can board at v_i and disembark at v_j for $1 \leq i < j \leq k$, and the reduced travel time for such passengers is $\sum_{a=i}^{j-1} c_{v_a v_{a+1}}$. Therefore, the goal is to operate the express train, determine the route and the passengers to board, with a limit of carrying only U passengers simultaneously, to maximize the total reduction in travel time for the boarded passengers.

The total reduction in travel time for the passengers in the express train scenario can be calculated by subtracting the total travel time with the express train from that without the use of express train. You can imagine that such situations will arise when determining the routes of subways and buses, installing highways, or setting up stations for returning trains.

sMTP is strongly NP-hard, as we would demonstrate in Chapter 3. In this thesis, we have conducted theoretical approaches on the polyhedral structure of the corresponding polyhedron and the approximation algorithm.

1.1 Related Problems

Backhaul Profit Maximization Problem

From the perspective of a freighter, the problem most similar to the sMTP is the Backhaul Profit Maximization Problem (BPMP) (Yu & Dong 2013, Dong et al. 2022). BPMP arises in the situation of third-party logistics providers in the maritime cargo transportation industry. They allow the freight carrier to deviate from its route during the backhaul process after reaching its original destination, in order to satisfy some of the other transportation demands and generate income. Therefore, like the sMTP, there are start and end nodes, it is not necessary to

visit all nodes, and the objective is maximizing profit from the voyage. Yu & Dong (2013) first proposed a mixed integer linear programming (MILP) model using arc and flow variables for the BPMP and suggested an exact solution procedure. Dong et al. (2022) presented another MILP formulation using triple variables and proposed a heuristic algorithm to obtain near-optimal solutions using this formulation. The BPMP can be considered a generalization of the sMTP, as it assumes that $p(k, l)$ is proportional to c_{kl}, there is an upper bound on the travel distance, and it assumes the unsplittable demand, meaning that either all or none of the demand (k, l) should be transported. However, research on this topic is limited to the formulating and simple heuristics. Research results on the polyhedral structure or approximation algorithm of the sMTP could also be helpful for the BPMP.

Line Planning Problem

The Line Planning Problem (LPP) is a problem in network design that simultaneously considers the routes that vehicles will pass through and how vehicles will travel on each route (Borndörfer et al. 2007). Depending on the situation, the problem can have many options in the modeling process, such as the possibility of transfers between multiple routes, setting the objective function, and selecting candidate routes (Schöbel & Scholl 2006, Schöbel 2012, Borndörfer et al. 2008). sMTP appears as a pricing subproblem when applying a column generation approach to LPP (Park et al. 2013). The column generation approach repeatedly searches for configurations that are effective routes in the LPP case when added to a candidate set of variables. It is motivated by the problem of finding the most helpful halting patterns for express trains in the existing system (Borndörfer et al. 2007). approached column generation for LPP in a different way. LPP is a highly complicated problem, and

it is necessary to compromise in various ways. In Schöbel (2012), a strategy is proposed that focuses on a limited number of candidates, rather than considering all possible paths in the underlying graph as potential routes. From this perspective, sMTP compromises by considering paths in an underlying graph in situations where the graph is acyclic. Although it is undoubtedly easier than a general problem, it is still a challenge, given that the number of possible paths can be exponentially large.

Traveling Repairman Problem with Profit

sMTP is related to the Traveling Repairman Problem with profit (TRPP) in that the objective function is formulated as profit minus cost, and not all nodes need to be visited. In TRPP, a constant-speed vehicle departs from a fixed root node and visits each node, earning a profit of $r_i - t_i$ if it visits node i at time t_i. If the graph is acyclic, this can be seen as a special case of sMTP. Coene & Spieksma (2008) presented a dynamic programming-based polynomial-time algorithm for TRPP when the nodes of the graph are on a straight line. Other studies on TRPP, such as Dewilde et al. (2013), have used heuristic approaches, including tabu search and greedy randomized adaptive search. Attempts have also been made using a hybrid evolutionary search algorithm and a general variable neighborhood search (Lu et al. 2019, Pei et al. 2020). Therefore, no research has been conducted on the polyhedral structure or approximation algorithms for this problem.

Traveling Salesman Problem with Profits

Problems called Traveling Salesman Problem with Profits (TSPP) are a type of vehicle routing problem that takes profits into consideration. While the Traveling Salesman Problem (TSP) aims to find the shortest distance tour that visits all nodes, TSPP considers situations where visiting some nodes may not be profitable or where

time constraints prevent all nodes from being visited (Feillet et al. 2005).

In TSPP, in addition to considering the distance traveled, rewards obtained from visited nodes or penalties obtained from unvisited nodes are also taken into account. There are various ways to approach this multi-objective optimization problem. For example, Balas (1989) studies a TSPP problem that aims to minimize the sum of the total distance traveled and the penalties associated with unvisited nodes and investigates valid inequalities for the corresponding polyhedron.

Another way to model TSPP is to use one of the two objectives as the objective function and the other as a constraint. A representative example of this is the orienteering problem, which aims to visit as many nodes as possible within a limited travel distance. Studies on the orienteering problem have investigated approximation algorithms such as those in Blum et al. (2007) and Chekuri et al. (2012), as well as various exact solution approaches and (meta)heuristics (Vansteenwegen et al. 2011).

TSPP differs from sMTP in that, instead of transporting goods from one node to another for revenue, a reward is obtained from just visiting each node, and cost is proportional to the distance traveled and not related to the amount of goods transported.

Dial-a-Ride Problem

sMTP simultaneously determines routes and profit-maximizing trades. With each trade having a different origin-destination pair, it can be compared to the Dial-a-Ride Problem (DaRP). The Dial-a-Ride service provides flexible transportation services for elderly or disabled individuals who cannot be covered by conventional public transportation, transporting them from their origin to their destination. A similar problem is Ridesharing (Furuhata et al. 2013), which can be seen to include

DaRP semantically, but primarily focuses on researching the profit gained when sharing the same vehicle from the passenger's perspective (Molenbruch et al. 2017, Ho et al. 2018). As survey papers Molenbruch et al. (2017) and Ho et al. (2018) show, similar diverse problems arise in various industries.

Most studies on DaRP assume that all demands must be satisfied. This seems to be because it is a service targeting the elderly and disabled who may have difficulty using other existing modes of transportation. Exceptionally, Parragh et al. (2015) and Jafari et al. (2016) considered the problem of selecting some of the requests among the requests. The service quality for each demand is reflected by constraints such as an upper bound on the ride time for each ride, known as the ride time constraint, or the time-window constraint, which requires both the arrival and departure times of a demand must contained in given time interval. Theoretical perspective on DaRP, in particular, as mentioned in Ho et al. (2018), research on the strength of the valid inequalities and the development of approximation algorithms are limited, despite the existence of research on valid inequalities in Cordeau (2006). This implies that research on sMTP's polyhedral structure and approximation algorithms could aid in the study of DaRP.

1.2 Objectives and contributions

Our objective is twofold. Firstly, we present two Mixed Integer Linear Programming (MILP) models for the sMTP, called Arc-Flow Formulation (AF) and Triple Formulation (TF), in which we consider binary variables y, the indicator vector of set of arcs used in a path, and flow and triple variables to represent trade volume. We then propose new families of valid inequalities for each formulation, and provide

conditions to be facet defining inequalities, and separation algorithms for some of these valid inequalities. Finally, we compare the gap between the objective value of integer solution and the fractional solution of the formulations with and without the valid inequalities.

Secondly, we study the approximability of the sMTP. We obtain a lower bound on the approximation ratio of the sMTP's approximation algorithm, and further obtain a lower bound on the approximation ratio of the approximation algorithm that utilizes the LP-relaxation of the MILP formulations. We also present an approximation algorithm that provides an upper bound on the ratio of approximation algorithms, while proposing an approximation algorithm for a special case of the sMTP with a single supply node and t-separable sMTP (with a better ratio than the approximation algorithm for sMTP). We then apply the techniques used here to present other cases, including the Traveling Repairman Problem with Profits, for which an approximation algorithm was previously unknown.

The followings are the contributions of the dissertation:

1. Polyhedral study on sMTP

    - We provide two new formulations.
    - We propose several families of valid inequalities for each formulation.
    - We propose the facet-defining condition and separation algorithm of each families of valid inequalities.
    - We conduct a computational experiment to assess the effectiveness of the proposed inequalities in improving the tightness of the LP formulation and reducing the computation time required for cut and branch by

computational experiment.

2. Approximability of sMTP

   - We provide the integrality gap of the formulations.
   - We provide inapproximability, a lower bound on approximation ratio of any polynomial-time algorithm for sMTP.
   - We devise approximation algorithms for the special cases of sMTP. They used as subroutine of the approximation algorithm for sMTP.
   - We devise an approximation algorithm for sMTP. Also we design the approximation algorithms for the problems related to sMTP, by applying the ideas used in the algorithm for sMTP.
   - We modify the algorithm to make it more practical and compare its performance with that of two other naive heuristic algorithms.

This thesis is composed of four chapters. In Chapter 2, two MILP formulations, named AF and TF are provided. Polyhedral structure of each formulation are studied, including set of valid inequality, conditions that the inequality becomes facet-defining, and corresponding separation algorithms. In Chapter 3, inapproximability result for sMTP, and approximation algorithms for sMTP and other related problems are presented. Finally in Chapter 5, summarized results and future research directions are presented.

Chapter 2

Polyhedral Study

In this chapter, we study the polyhedral structure of sMTP. We introduce two formulations called Arc-Flow Formulation and Triple Formulation and study the valid inequality for each related integer hull. For the Arc-Flow formulation, we derive a family of valid inequalities, identify their facet-defining conditions, and present an $O(n^4)$-time separation algorithm, which is same as the number of variables. Based on this result, we propose generalized families of valid inequalities for AF. For the Triple formulation, we investigate their valid inequality in a different manner. We conduct an experiment to compare the bounds obtained from the LP relaxation of the formulations with or without adding the valid inequalities.

2.1 Introduction

Many of combinatorial optimization problem can be formulated as follow:

$$\begin{aligned} \max \quad & c^T x + d^T y, \\ \text{subject to} \quad & Ax + By \leq b, \\ & x \in \mathbb{R}^n, y \in \mathbb{Z}^m. \end{aligned}$$

Where A and B are rational matrices, b, c, and d are rational vectors. Let P_{int} be $conv\{(x, y)|Ax + By \leq b, x \in \mathbb{R}^n, y \in \mathbb{Z}^m\}$, the set of vectors that can be respresented as a convex combination of feasible solutions, and P be $\{(x, y)|Ax + By \leq b, x \in \mathbb{R}^n, y \in \mathbb{R}^m\}$. If $P_{int} = P$, the LP-rexalation problem, maximize the objective function on P can be solved fast practically or theoretically using the simplex method or the ellipsoid method, the optimal solution, which is a vertex of P, can be obtained and it satisfies the integer condition of y, thus yielding the optimal solution of the original optimization problem (Grötschel et al. 2012).

In general, P does not guarantee that the y-coefficients of its vertices are integers, so it is not the same as P_{int}, and the solution obtained by solving the LP-relaxation does not guarantee to be a feasible solution of the original problem. Instead, the objective function value obtained by solving the LP-relaxation becomes an upper bound of the optimal objective function value because the integer constraints are relaxed, increasing the feasible solution set. On the other hand, the obtained solution can provide a lower bound of the objective function from which a good quality feasible solution may be found nearby. The upper and lower bounds obtained from this method can be utilized in various methods, including branch-and-bound. When using these methods, finding a description of P that is closer to P_{int} provides an opportunity to obtain better bounds and a better solution. An inequality of the form $a^T x + b^T y \leq k$ that all elements of P_{int} satisfies is called a valid inequality, and ideally, a valid inequality that is necessary to describe P_{int} is called a facet-defining inequality.

In this chapter, our aim is to find valid inequalities and facet-defining conditions in two formulations, the Arc-Flow formulation and the Triple formulation. Addi-

tionally, if there are so many valid inequalities, usually a procedure to determine whether a given vector satisfies all inequalities or to identify a violated inequalities is needed. This procedure is called a separation algorithm, and our research also aims to find the corresponding separation algorithms for the valid inequalities found. In the final part of this section, we introduce the Arc-Flow Formulation (AF), review the relevant literature, and summarize the unsuccessful results obtained from other integer programming approaches.

2.1.1 Arc-Flow Formulation

The Arc-Flow Formulation (AF) is a straightforward mixed integer programming formulation. For $1 \leq k \leq l \leq n$, let the *trade variable* x_{kl} be, as before, the volume of the products supplied from node k to node l. Also denote by the *flow variable* f_{ij}^{kl} for $k \leq i < j \leq l$, the volume of the product from k to l through arc ij. Finally, for each arc ij with $1 \leq i < j \leq n$ define the binary *path variable* y_{ij} which indicates whether arc ij is used in the path or not. Note that we may assume $d(k,l) \leq U$ for all $1 \leq k < l \leq n$. Also we will assume $d(k,l) > 0 \ \forall \ 1 \leq k < l \leq n$. This is not a restriction, from that for otherwise we can reassign to (k,l) $d(k,l) = U$ and $r(k,l)$ $= 0$ so that $x_{kl} = 0$. Since x_{kl} can be expressed as f-variables in constraints (2.3) with $i = k$, an equivalent system without x_{kl} can be represented, but we use them for readability. The (AF) is formulated as an MILP as followings.

Problem 2.1.1. Arc-Flow Formulation (AF)

$$\max \quad \sum_{1\le k<l\le n}\left(r(k,l)x_{kl} - \sum_{k\le i<j\le l} c_{ij} f_{ij}^{kl}\right) \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j:i<j\le n} y_{ij} - \sum_{j:1\le j<i} y_{ji} = \begin{cases} 1, & i=1 \\ 0, & 1<i<n \end{cases}, \tag{2.2}$$

$$\sum_{j:i<j\le l} f_{ij}^{kl} - \sum_{j:k\le j<i} f_{ji}^{kl} = \begin{cases} x_{kl}, & i=k \\ 0, & k<i<l \end{cases}, \quad 1\le k\le i<l\le n, \tag{2.3}$$

$$\sum_{k\le i<j\le l} f_{ij}^{kl} \le U y_{ij}, \qquad 1\le i<j\le n, \tag{2.4}$$

$$x_{kl} \le d(k,l), \qquad 1\le k<l\le n, \tag{2.5}$$

$$y_{ij}\in\{0,1\},\ x_{kl}\ge 0,\ f_{ij}^{kl}\ge 0, \qquad 1\le k\le i<j\le l\le n. \tag{2.6}$$

The constraints (2.2) enforce the arcs (i,j)'s with $y_{ij}=1$ constitutes a 1-n path. The constraints (2.3) conserve the flow on the subgraph induced by the nodes $\{k,k+1,\ldots,l\}$ so that the volume x_{kl} is sent from k to l. (2.5) are the demand constraints. They can be tightened if replaced with the constraints

$$f_{ij}^{kl} \le d(k,l) y_{ij},\ 1\le k<l\le n,\ k\le i<j\le l. \tag{2.5$'$}$$

It is not difficult to see that the replacement yields a stronger formulation of sMTP. We will refer to (2.1-2.4), (2.5$'$), and (2.6) as the *Arc-Flow Formulation (AF)*. Let the relaxed polytope defined by the constraint expression (AF) be denoted by $P(AF)$, and its integer hull be denoted by $P_{int}(AF)$. When considering (AF) explicitly, we may also represent them as P and P_{int}, respectively.

2.1.2 Relationships with Other Formulations

In a general graph, the constraint that the characteristic vector of the set of arcs must form a single path has been extensively studied in various research, including the Traveling Salesman Problem (TSP) (Goemans 1995, Balas 1989, An et al. 2015). In fact, this is a polyhedral study of the s-t path TSP problem, where we seek the starting and ending nodes are fixed and the shortest path that visits all nodes exactly once. However, in the case of sMTP, since an acyclic graph is considered, the path constraint in the polyhedron is sufficient with (2.2), from that the coefficient matrix of the constraint is a totally unimodular matrix. On the other hand, with the exception of path constraints, it becomes similar with network design problem. The typical formulation for the multicommodity capacitated network design problem is expressed as follows. This is a substructure that frequently appears in the formulation of various other problems (Magnanti & Wong 1984).

$$\sum_j f_{ij}^{kl} - \sum_j f_{ji}^{kl} = \begin{cases} d_{kl}, & i = k \\ 0, & i \neq k, l \end{cases}, \quad k, i, l \in N, \tag{2.7}$$

$$\sum_{k \leq i < j \leq l} f_{ij}^{kl} \leq U_{ij} y_{ij}, \quad k, i, j, l \in N, \tag{2.8}$$

$$y_{ij} \in \mathbf{Z}_0^+, \ f_{ij}^{kl} \geq 0, \quad k, i, j, l \in N. \tag{2.9}$$

Denote the polyhedron consisting of such constraints as Network Design Formulation (NDF). In this problem, the volume of flow that must pass between pairs of vertices is predetermined. If a facility is installed in an arc, the amount of flow that can pass through that arc increases. The goal is usually to minimize the cost

of installing facilities while ensuring that all flows can pass through by installed facilities. There can be various conditions. For example, the constraint varies to the underlying graph is whether undirected or directed graph.

sMTP's Arc-Flow Formulation (AF), which we introduce, uses the f and y variables in the formulation. The difference between (NDF) and the structure of sMTP excluding the path constraint is that x_{kl} becomes a constant d_{kl}. In addition, sMTP only considers cases where U_{ij}'s are identical for ij. When U_{ij}'s are identical, it is commonly referred to as the Network Loading Problem (Agarwal 2018). When substituting x_{kl} for d_{kl} in the valid inequality of (NDF), there may be cases where it becomes a valid inequality of (AF), but there may also be cases where it does not. One of the typical examples is the class of valid inequalities that apply the cover inequality of the knapsack problem in (NDF) (Chouman et al. 2017). The cover inequality refers to an equation in the form of $\sum_{ij \in S} y_{ij} \geq k$ or $\sum U_{ij} y_{ij} \geq D$, where S, k, and D are some function values, including rounding for d_{kl}, in order to satisfy the flow by collecting some demand pairs to meet the obligatory demand. If we expressed the coefficient as x, the inequality might be nonlinear.

The (NDF) has a large number of constraints and variables. Consequently, attempts to reduce the size of the formulation have been made, even if they have to sacrifice tight bounds. One such attempt is to treat flows that originate from the same node as the same commodity. In this case, if we let δ_i^l denote the amount of commodity that departs from node i and arrives at node l, then d_{il} is equal to $-\sum_{l>j} d_{jl}$ for $i = l$, 0 for $i > l$, and d_{il} for $i < l$. Also, we denote u_{ij}^l, the aggregated flow represents the flow passing through arc ij that ends at node l, then (2.7) can be expressed as follows:

$$\sum_i u_{ij}^l - \sum_i u_{ji}^l = \delta_j^l, \forall i, l \in N$$

This method can be applied even when demand is non-obligatory, similar to sMTP, and we introduce the formulation that uses u and y as variables, named Triple Formulation (TF). Although it uses fewer variables, it generally provides worse relaxation bounds than (AF).

Research on the valid inequalities of NDP includes studies on the generalizations of flow cover inequalities. The flow cover inequality is a formula that relates the balance of a vertex, the capacities of incoming and outgoing arcs from other vertices, and is considered as a valid inequality (Gu et al. 1999). While this inequality can be aggregated to obtain a valid inequality for single-commodity by considering the problem of multi-commodity, it is not strong enough, so additional lifting procedures are required.

Efforts have been made to generalize this inequality, which has the advantage and disadvantage of being simple. Atamtürk et al. (2016) and Atamtürk et al. (2017) extended the flow cover inequality that could be considered as a relationship between one vertex and another vertex to the relationship between three vertices and the relationship between vertices that constitute a path, respectively. Wolsey (1989) found a broad class of valid inequalities using submodularity.

However, when applying these inequalities to sMTP, many redundant inequalities are obtained due to the fact that the capacities of the arcs are the same and the graph is acyclic. Therefore, in order to correspond to the sMTP, including (AF) and (TF), we need to focus on finding facet-defining inequalities of (NDF).

Like sMTP, problems with the assumption that the form of the network installed in NDF should have a specific shape such as tree or ring have been considered (Chekuri et al. 2013, Lee et al. 2009). This is because when the installation cost of the network is high, a tree is the best option to ensure connectivity for all nodes, and a ring is the best option when maintaining connectivity even if a facility corresponding to a certain edge fails, such as in 2-connectivity. There are also studies on problems that consider the condition that the flow between each o-d pair must form a single path.

Budget Design Problem

Wong (1980) considers the Budget Design Problem, which involves a budget constraint in the Network Design Problem (NDP), i.e., the network installation cost is not considered but a budget for installation is given. The budget constraint is represented in the form of $\sum q_{ij} y_{ij} \leq B$, which sufficiently captures the condition that y must be a path from 1 to n on an acyclic graph. Wong (1980) showed that this problem is $n^{1-\varepsilon}$-hard to approximate for any positive ε, using a reduction from the Steiner Tree problem, while also presenting a $2n$-approximation algorithm. However, no further research has been conducted on this problem.

Unsplittable Flow Problem

The Unsplittable flow problem is a problem with an additional constraint that the flow for each $o-d$ pair cannot be split into multiple paths, given the situation of NDP described above. This can be viewed as a problem closer to the sMTP, where all pairs must flow within a single path, but it is difficult to design a formulation for it. One simple approach is to use binary variables to indicate which arcs are used for each $o-d$ pair, but this generates a number of binary variables that is proportional

to the product of the number of pairs and arcs. Another approach is to create binary variables that are equal to 1 when a certain path is used for each possible path for each $o-d$ pair. A typical compromise in solving this problem is to limit the number of possible paths to a polynomial number, which is weaker than the assumption that the underlying graph is acyclic.

On the other hand, many studies have been conducted on approximation algorithms for this problem, such as Chakrabarti et al. (2007), Chekuri et al. (2006). This problem includes all-or-nothing multicommodity flow problems, edge disjoint path problems, and unsplittable flow problems on paths, and many studies have been conducted on approximation algorithms for these problems (Kawarabayashi et al. 2012, Chuzhoy & Li 2016, Seguin-Charbonneau & Shepherd 2021, Chekuri & Khanna 2007, Chuzhoy & Kim 2015). These algorithms mostly have a ratio of $O(n^c)$ unless assuming special graphs such as trees or paths, and research has been conducted on the corresponding hardness of approximation (Ma & Wang 2000, Guruswami et al. 2003, Andrews et al. 2005).

2.1.3 Other Methods

In this subsection, we aim to summarize alternative approaches rather than finding valid inequalities for sMTP. Firstly, we can consider the Danzig-Wolfe decomposition using path formulation. A path formulation can be represented as follows.

$$
\begin{aligned}
\max \quad & \sum_{1\le k<l\le n} r(k,l) \sum_{p\in\mathcal{P}^{kl}} f_p^{kl} - \sum_{1\le k<l\le n} \sum_{p\in\mathcal{P}^{kl}} c^{kl}(p) f_p^{kl} \\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}} y_p = 1, \\
& \sum_{p\in\mathcal{P}_{ij}} \sum_{k\le i,j\le l} f_p^{kl} \le U \sum_{p\in\mathcal{P}_{ij}} y_p, && ij\in A, \\
& \sum_{p\in\mathcal{P}^{kl}:ij\in A(p)} f_p^{kl} \le d(k,l) \sum_{p\in\mathcal{P}^{kl}:ij\in A(p)} y_p, && 1\le k\le i<j\le l\le n, \\
& y_p\in\{0,1\},\ x_p^{kl}\ge 0, && p\in\mathcal{P},\ 1\le k<l\le n.
\end{aligned}
$$

$\mathcal{P}$ represents the set of all 1-n paths. $\mathcal{P}_{ij}$ represents the set of paths in $\mathcal{P}$ that pass through arc ij. $\mathcal{P}^{kl}$ represents the set of paths in $\mathcal{P}$ that pass through nodes k and l. y_p is a binary variable that equals 1 if path p is used and 0 otherwise. f_p^{kl} represents the value of x_{kl} when path p is used. The following are the subproblems that arise when applying the Danzig-Wolfe decomposition to the given problem:

$$
\begin{aligned}
\max \quad & \sum_{ij\in A(p)} \left(\sum_{k\le i,j\le l} d(k,l)\gamma_{ij}^{kl} + U\mu_{ij} \right) - \alpha \\
\text{s.t.} \quad & p\in\mathcal{P}.
\end{aligned}
$$

Even when all μ_{ij} are equal to zero, it becomes a sMTP with no capacity constraints, where for every pair of nodes (k,l) traversed by a path, exactly $d(k,l)$ trades should be conducted.

The remainder of this chapter is organized as follows. In Section 2.2, we analyze the dimension of the polyhedron corresponding to (AF) and the necessary conditions for the satisfied valid inequality obtained by projecting from the extended formula-

tion. In Section 2.3, we propose families of valid inequalities for (AF). Firstly, we present the 3-Criteria Inequality, along with the conditions for being a facet-defining inequality and the separation algorithm. Then, we propose generalized 3-Criteria Inequality to other families of valid inequalities and suggest conditions for them to be facet-defining valid inequality. In Section 2.4, we introduce the Triple Formulation, analyze the basic property, and propose families of valid inequalities.

2.2 Basic Properties of Arc-Flow Formulation

In this section, we present the basic properties of Arc-Flow Formulation (AF), a condition of an inequality to be a valid inequality and the dimension of polyhedron.

2.2.1 Necessary Conditions of Facet-defining Inequality

We can obtain a exact extended formulation by introducing binary variables path variables, z_p, which is 1 if and only if using and 1-n path p, as well as z_p^{kl}, which is the amount of k-l flow if and only if using and 1-n path p and 0 otherwise, and project this formulation onto the (y, f)-space to obtain the conditions for a valid inequality. Let $\mathcal{P}$ be the set of all 1-n paths, where each path is considered as a set of nodes and arcs.

Theorem 2.2.1. *In (y, f)-space, the only form of facet-defining inequality, apart from the constraints of (AF), is as follows:*

$$\sum a_{ij} y_{ij} \geq \sum b_{ij}^{kl} f_{ij}^{kl} - \sum c_{ij}^{kl} f_{ij}^{kl} \ \ (a, b, c \geq 0)$$

The necessary condition for this inequality to be a facet-defining inequality and the sufficient condition for it to be a valid inequality is that the following holds for

each $p \in \mathcal{P}$*:*

$$\sum_{ij \in p} a_{ij} \geq \max_{\mu_i \geq 0}(U \sum_{i=1}^{n-1} \mu_i + \sum_{k,l \in p} d(k,l) \max\{\sum_{ij \in p} (b_{ij}^{kl} - c_{ij}^{kl}) - \sum_{i=k}^{l-1} \mu_i, 0\})$$

Proof: We will consider a path formulation, which is an exact formulation, and project it onto the (y, x, f)-space to obtain the above result. In the (y, x, f) space of the integer hull $P_{int}(AF)$ of the sMTP problem, the y_{ij}'s for $1 \leq i < j \leq n$ are integers, and hence they constitute a single path. The set of all vertices of $P_{int}(AF)$ can be partitioned into those that form 1-n paths with the same y values. A necessary and sufficient condition for a given (y, x, f) to be an element of $P_{int}(AF)$ is that it can be expressed as a convex combination of the vertices of that path. We can consider this as a convex combination of multiple convex combinations of vertices that have the same y values. Maintaining the integer value of y within these combinations, being an element of $P_{int}(AF)$ is essentially being expressible as a convex combination of the integer solutions of 2^{n-2} distinct 1-n paths. By integrating this concept into the constraints, we can derive an exact formulation of $P_{int}(AF)$.

A given (y, x, f) is an element of P_{int} if and only if there exists a set of integer feasible solutions $(y_p, x_p, f_p), p \in \mathcal{P}$ that satisfies the following conditions.

$$(y, x, f) = \sum_{p \in \mathcal{P}} z_p(y_p, x_p, f_p), \sum_{p \in P} z_p = 1, \ (y_p, x_p, f_p) \in P_{int}.$$

y_p is the arc characteristic vector of p. Therefore, it is a constant, and comparing the y components on both sides yields a linear equation in terms of y and z_p. In order to make the equation comparing the coefficients of x on both sides a linear equation as well, a new variable $z_p^{kl} = z_p \pi_{kl}(x_p)$ is introduced. The equations comparing the coefficients of y and x are then as follows.

$$\sum_{p\in\mathcal{P}:ij\in\mathcal{P}} z_p = y_{ij}, \qquad 1 \le i < j \le n, \tag{2.10a}$$

$$\sum_{p\in\mathcal{P}:ij\in p,k,l\in p} z_p^{kl} = f_{ij}^{kl}, \qquad 1 \le k \le i < j \le l \le n, \tag{2.10b}$$

$$z_p^{kl} \le d(k,l) z_p, \qquad 1 \le k < l \le n, p \in \mathcal{P}, \tag{2.10c}$$

$$\sum_{p\in\mathcal{P},k,l:1\le k\le i,j\le l\le n,k,l,ij\in p} z_p^{kl} \le U y_{ij}, \qquad 1 \le i < j \le n, \tag{2.10d}$$

$$z_p, z_p^{kl} \ge 0, \qquad 1 \le k < l \le n, p \in \mathcal{P}. \tag{2.10e}$$

(2.10a) corresponds to (2.2), (2.10b) corresponds to (2.3), and (2.10c) corre sponds to (2.5). (2.10e) corresponds to (2.6). Since $f_{ij}^{kl} = y_{ij}x^{kl}$, the coefficients of f do not need to be compared. The equal sign in (2.10a) can be changed to $\le$ without loss of generality. If there exists a slack, the slack forms collection of paths from 1 to n, and can be represented as a conical combination of feasible solutions with $f = 0$. Since this is an exact formulation, we will now project it onto the (y, x, f)-space to obtain a necessary condition and sufficient condition for an inequality to be a facet-defining inequality of (AF).

From $\sum z_p = 1$, we obtain (2.2). By Farkas' Lemma, it is known that if $P = \{(z, w) : Az \le Bw, z, w \ge 0\}$, then $\pi_w P = \{w | v^T \ge 0, v^T A \ge 0 \Rightarrow v^T Bw \ge 0\}$. Here, $\pi_w P$ denotes the projection of P onto the w-space. In this problem, if we let $z = (z_p^{kl}, z_p)$ and $w = (f, y)$, the constraint $Az \le Bw$ is as follows. The non-negativity constraint is omitted.

$$
\begin{aligned}
-\sum_{p\in P: ij\in p, k,l\in p} z_p^{kl} &\leq -f_{ij}^{kl}, & 1\leq k\leq i<j\leq l\leq n, \quad (2.11a)\\
\sum_{p\in P: ij\in p, k,l\in p} z_p^{kl} &\leq f_{ij}^{kl}, & 1\leq k\leq i<j\leq l\leq n, \quad (2.11b)\\
z_p^{kl} - d(k,l)z_p &\leq 0, & 1\leq k<l\leq n, r(k,l)\in P \quad (2.11c)\\
\sum_{k,l:1\leq k\leq i<l\leq n, k,l\in p} z_p^{kl} - Uz_p &\leq 0, & 1\leq i<n, p\in\mathcal{P} \quad (2.11d)\\
\sum_{p\in P: ij\in p} z_p &\leq y_{ij}, & 1\leq i<j\leq n. \quad (2.11e)
\end{aligned}
$$

Now let us apply the projection above. Let A be the coefficient matrix of the left-hand side, and consider v such that $v^T \geq 0$ and $v^T A \geq 0$. The inequality obtained from v with only the coefficients of (2.11a) and (2.11b) being nonzero is (2.3). Excluding the nonnegativity constraint, we can see that it is sufficient to consider inequalities of the following form here.

$$\sum a_{ij}y_{ij} \geq \sum b_{ij}^{kl} f_{ij}^{kl} - \sum c_{ij}^{kl} f_{ij}^{kl} \ (a,b,c\geq 0)$$

Assume that b and c are fixed. Then, the z_p^{kl} coefficients with negative values should be added by conical combinations of (2.11c) or (2.11d) so that the coefficients become non-negative. Furthermore, both (2.11c) and (2.11d) decrease the coefficient of z_p, which should be reduced by conical combinations of (2.11e).

We can say that both (2.11c) and (2.11d) play a role in transferring the negative coefficient of z_p^{kl} to the negative coefficient of z_p. Given p, if the coefficient of (2.11d) is μ_i for each $1\leq i<n$, the coefficient of z_p^{kl} increases by $\sum_{k\leq i<l} \mu_i$. The remaining negative coefficients of z_p^{kl} can be precisely converted into coefficients of z_p increased

by a factor of $d(k,l)$ using (2.11c). Therefore, we have shown the first statement of the theorem. If (2.11d) does not exist, it becomes simpler as in the second statement. $\square$

Although we obtained a necessary condition for a facet-defining inequality in Theorem 2.2.1, we may want to find a condition that is sufficient. It was conjectured that if v_0 belongs to an extreme ray of the polyhedron $\{v|v \geq 0, v^T A \geq 0\}$ that appears in the projection, then $v_0^T By \geq 0$ could be a facet-defining inequality, but this has been shown to be not always true (Balas 1998). Checking whether a given inequality is a valid inequality for a general integer program is a problem belonging to co-NP, and checking whether it is a facet-defining inequality is known to be a problem belonging to a class called D^p, which includes both NP and co-NP (Papadimitriou & Yannakakis 1982).

Now, we compute the dimension of this formulation. We can show that this is equal to the difference between the number of variables and the number of equations.

Proposition 2.2.2. $\dim(P(AF)) = (n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24}$.

Proof:

First, we show that the left-hand side is less than or equal to the right-hand side. There are $\binom{n}{2}$ trade variables x_{kl} and $\binom{n}{2}$ path variables y_{ij}. To each trade variable x_{kl}, the number of corresponding flow variables f_{ij}^{kl} is $\binom{l-k+1}{2}$. The total number of flow variables is $(n-1)\binom{2}{2} + (n-2)\binom{3}{2} + \cdots + 1\binom{n}{2} = \frac{(n-1)n(n+1)(n+2)}{24}$. The rank of (2.2) is $n-1$. The rank of (2.3) is $1(n-1) + 2(n-2) + \cdots + (n-1)(n-(n-1)) =$

$\frac{n(n+1)(n-1)}{6}$. Thus the total rank is $n-1+\frac{n(n+1)(n-1)}{6}$. Therefore,

$$\begin{aligned} \dim P &\leq 2\left(\binom{n}{2}\right) + \frac{(n-1)n(n+1)(n+2)}{24} - (n-1) - \frac{n(n+1)(n-1)}{6} \\ &= (n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24}. \end{aligned} \tag{2.12}$$

Now we show that the inequality in the opposite direction also holds. We can find $(n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24} + 1$ affinely independent integer vectors, where the integer solution satisfies $f_{ij}^{kl} = x_{kl}y_{ij}$, so we only represent x and y. Any variables not represented are all equal to 0.

- y_{1n}: $y_{1n} = 1$
- y_{1k}: $y_{1k}, y_{kn} = 1$ (for k such that $1 < k < n$)
- y_{kl}: $y_{1k}, y_{kl}, y_{ln} = 1$ (for k, l such that $1 < k < l < n$)
- x_{kl}: $y_{1k}, y_{kl}, y_{ln} = 1$, $x_{kl} = 1$ (for k, l such that $1 < k < l < n$)
- f_{ki}^{kl}: $y_{1k}, y_{ki}, y_{il}, y_{ln} = 1$, $x_{kl} = 1$ (for k, l such that $1 < k < l < n$)
- f_{ij}^{kl}: $y_{1k}, y_{ki}, y_{ij}, y_{jl}, y_{ln} = 1$, $x_{kl} = 1$ (for k, l such that $1 < k < l < n$)

When considering them in order from top to bottom, there exist variables that were continuously 0 and newly became 1, namely y_{kn} with $1 < k < n$ and f_{il}^{kl} with $1 \leq k \leq i < l \leq n$. These variables correspond one-to-one with equations in (AF), so we obtain $\dim P(AF) = (n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24}$. $\square$

2.3 Valid Inequalities for the Arc-Flow Formulation

In this section, we aim to find valid inequalities for the Arc-Flow Formulation (AF).

2.3.1 Facet-defining Inequalities: 3-Criteria Inequality

3-Criteria Inequality

We now introduce the set of valid inequalities called 3-criteria inequality. Let three a, b, and c satisfying $a < c \leq b$. Define $V_1 = \{i \in V | 1 \leq i < a\}$, $V_2 = \{i \in V | a \leq i < c\}$, $V_3 = \{i \in V | c \leq i \leq b\}$, and $V_4 = \{i \in V | b < i \leq n\}$. Also for each pair $p, q \in \{1, 2, 3, 4\}$ with $p < q$, let $A_{pq} = \{ij \in A | i \in V_p, j \in V_q\}$, $C^{pq} = \{(k, l) | k \in V_p, l \in V_q\}$. Suppose $\pi : A_{12} \to C^{13}$ be any function mapping arc ij to node pair (k, l) so that $k \leq i < j \leq l$. Also suppose $\phi : A_{34} \to C^{24}$ maps arc ij to node pair (k, l) so that $k \leq i < j \leq l$.

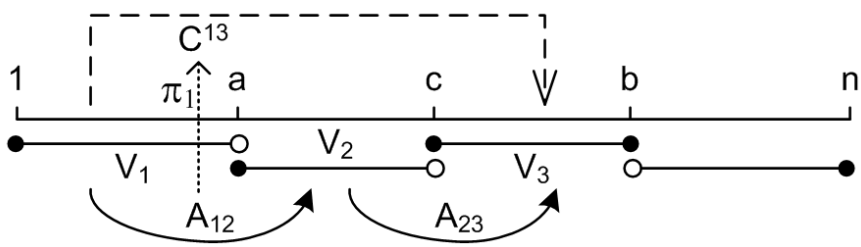

Figure 2.1: A_{pq}'s, C^{pq}'s, π.

Proposition 2.3.1. *The followings are valid inequalities of sMTP:*

$$\sum_{ij \in A_{12}} \frac{f_{ij}^{\pi(ij)}}{d(\pi(ij))} \leq \sum_{ij \in A_{23}} y_{ij} \tag{2.13}$$

$$\sum_{ij \in A_{34}} \frac{f_{ij}^{\phi(ij)}}{d(\phi(ij))} \leq \sum_{ij \in A_{23}} y_{ij}. \tag{2.14}$$

Proof Let (y, x, f) be any feasible solution of sMTP. A 1-n path can not use more than one arc from A_{23}: $\sum_{ij \in A_{23}} y_{ij} = 0$ or 1. Suppose $\sum_{ij \in A_{23}} y_{ij} = 0$. Then a path

can not use any arc from A_{23}. Since for $(k,l) \in C^{13}$ and for $ij \in A_{12}$, any flow f_{ij}^{kl} should be relayed along an arc from A_{23}, it implies $f_{ij}^{kl} = 0$ and hence (2.13) holds. The validity of (2.14) is similar and omitted. □

In fact, (2.13) and (2.14) define facets of $P_{int}(AF)$, the integer hull of the arc formulation of sMTP. Since the proofs for (2.13) and (2.14) are symmetric, we only provide the proof for (2.13). Let

$$H = \left\{ w := (y, x, f) : \sum_{ij \in A_{12}} \frac{f_{ij}^{\pi(ij)}}{d(\pi(ij))} = \sum_{ij \in A_{23}} y_{ij} \right\}. \tag{2.15}$$

Let F be the intersection of the hyperplane H and $P_{int}(AF)$. In further discussion, we denote $\pi(ij) = (\pi_1(ij), \pi_2(ij))$ and $\phi(ij) = (\phi_1(ij), \phi_2(ij))$.

Theorem 2.3.2. *The 3-Criteria inequalities, (2.13) defines a facet of $P_{int}(AF)$ other than (2.5′) if and only if*

1) $b \neq n$,

2) $\exists p \in V_1$ *such that* $\pi((a-1)a) = (p, b)$,

3) $\exists q, r \in V_1$ *such that* $\pi(ra) = (q, b)$ *and* $q \neq p$ *from 2),*

4) $\exists s, t \in V_1$ *such that* $\pi(ta) = (s, c)$, *and*

5) $U > \max_{ij \in A_{12}} \{d(\pi(ij))\}$.

Symmetrically, (2.14) defines a facet of $P_{int}(AF)$ other than (2.5′) if and only if

1) $a \neq 1$,

2) $\exists p' \in V_4$ *such that* $\phi(b(b+1)) = (a, p')$,

3) $\exists q', r' \in V_4$ *such that* $\phi 2(br') = (a, q')$ *and* $q' \neq p'$ *from 2),*

4) $\exists s', t' \in V_4$ *such that* $\phi(bt') = (c-1, s')$*, and*

5) $U > \max_{ij \in A_{34}} \{d(\phi(ij))\}$.

Proof: We provide the proof of the first part of the theorem only.

(**Necessity**) We first establish the necessity of the theorem. In doing so, we rely on the following two observations.

i) For each quadruple $k \leq i < j \leq l$, F contains a point $w = (y, x, f)$ with y integral and $f_{ij}^{kl} > 0$.

Proof of i): For any quadruple $k \leq i < j \leq l$, consider the face $G_{ij}^{kl} := \{x \in P_{int}(AF) : f_{ij}^{kl} = 0\}$. Then G_{ij}^{kl} is a proper face of $P_{int}(AF)$ since, for each $k \leq i < j \leq l$, any feasible solution using path $1 - k - i - j - l - n$ with $x_{kl} > 0$ belongs to $P_{int}(AF) \setminus F$.

Also the feasible solution using path traversing every node but no trade between any pair of nodes and hence having $f_{ij}^{kl} = 0$ satisfies (2.13) with strict inequality. Hence $F \neq G_{ij}^{kl}$ for every quadruple $k \leq i < j \leq l$. It means there is vertex of F not belong to G_{ij}^{kl} for every quadruple $k \leq i < j \leq l$. Thus i) follows.

ii) $(2.5')$ defines a proper face of $P_{int}(AF)$, i.e. a face which is a proper subset of $P_{int}(AF)$.

Proof of ii): For any $k \leq i < j \leq l$, a feasible solution using path $1 - i - j - n$ but no flow between nodes satisfies $(2.5')$ with strict inequality.

Consider the conditions sequentially.

1) Suppose on the contrary $b = n$. Then if we add the inequalities (2.5′) for all $ij \in A_{12}$ such that $(k, l) = \pi(ij)$, we get $\sum_{ij \in A_{12}} \frac{f^{\pi(ij)}}{d(\pi(ij))} \leq \sum_{ij \in A_{12}} y_{ij} = \sum_{ij \in A_{23}} y_{ij}$. Since the facet F is generated as a conic combination of inequalities (2.5′), it should be a facet defined by an inequality from (2.5′) contradicting the assumption of the theorem.

2) From 1), there is $l \in V_4$ such that $b < l$. Then applying i) to the quadruple $(a-1, a, b, l)$, there is a feasible solution $w = (y, x, f) \in F$ such that $f_{ab}^{(a-1)l} > 0$ and y is integral. In particular, $y_{ab} = 1$ and hence the path traverses nodes $a - 1$, a, b and no other node from V_3 than b as in Figure 2.2 a).

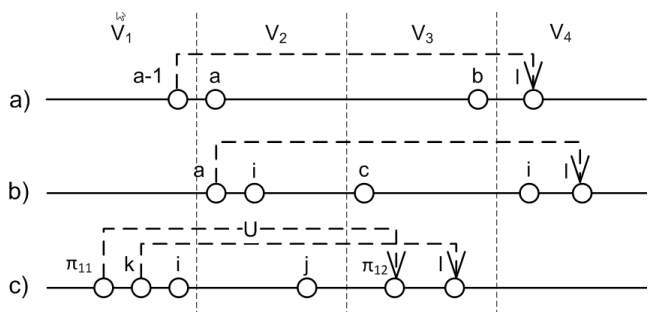

Figure 2.2: Solutions in the necessity proof

Since the path traverses the first node a of V_2 and only b from V_3, and $f_{ab}^{(a-1)l} > 0$, there should be $p \in V_1$ such that $f_{(a-1)a}^{pb} > 0$ in the left hand side of (2.13).

3) Suppose on the contrary $\pi_1(i_1 a)$ is p, for every $i_1 \in V_1$ satisfying $\pi_2(i_1 a) = b$. Consider any feasible solution $w = (y, x, f)$ lying on F. If $y_{ab} = 1$, then since $ab \in A_{23}$ there should be a unique $ij \in A_{12}$ such that $f_{ij}^{\pi(ij)} = d(\pi(ij))$ and the rest f-variables are 0 in the left hand side of (2.13). Since a (b, resp.) is the first (last,

resp.) node of V_2 (V_3, resp.), there should be unique q and r with $q \leq r < a$ such that $\frac{f_{ra}^{qb}}{d(k,b)} = 1$ in the left hand side of (2.13). From the assumption we have $q = p$ and therefore $\frac{f_{ra}^{pb}}{d(p,b)} = 1$. As $y_{ab} = 1$, we get $\frac{f_{ra}^{pb}}{d(p,b)} = \frac{f_{ab}^{pb}}{d(p,b)} = 1$. If on the other hand $y_{ab} = 0$, then $f_{ab}^{pb} = 0$ and $y_{ab} = \frac{f_{ra}^{pb}}{d(p,b)} = \frac{f_{ab}^{pb}}{d(p,b)}$. Thus w always satisfies (2.5′) for $(k, i, j, l) = (p, a, b, b)$ implying F is defined by an inequality from (2.5′) and contradicting to the assumption of the theorem.

4) Similarly with the proof of 2), i) guarantees a feasible solution on F in which $f_{cl}^{al} > 0$ for $l \in V_4$ and y is integral, which implies the existence of s, $t \in V_1$ such that $f_{ta}^{sc} > 0$ and hence $\pi_1(ta) = (s, c)$. See Figure 2.2 b).

5) Suppose, on the contrary, $U = d(\pi(ij))$ for some $ij \in A_{12}$. From 3), we have p, $q \in V_1$ and $p \neq q$, and hence $|V_1| \geq 2$. We can take $k \in V_1$ such that $k \neq \pi_1(ij)$. See Figure 2.2 c). Consider a feasible solution on F guaranteed by i) such that $f_{ij}^{k\pi_2(ij)} > 0$ and y is integral. Since the right hand side of (2.15) is 1 and $y_{ij} = 1$, $f_{ij}^{\pi(ij)} = s_{\pi(ij)} = U$. Thus on arc ij, the flows $f_{ij}^{k\pi_2(ij)}$ and $f_{ij}^{\pi(ij)}$ sum greater than U violating (2.4).

(**Sufficiency**) We now prove the sufficiency of the theorem. Let F' be a translation of F containing the origin and S the subspace defined by the linear hull of F'. Notice that $\dim F < \dim P_{int}$: the feasible solution obtained by letting $y_{1a} = y_{ab} = y_{bn} = 1$, and the rest of the variables equal to 0 belongs to $P_{int} \backslash F$. Hence by Proposition 2.2.2, to prove F is a facet, it suffices to find a set of independent vectors lying in S whose cardinality equals to $(n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24} - 1$. Recall we have $\binom{n}{2}$ x_{kl}'s, $\binom{n}{2}$ y_{ij}'s, and $\sum_{q=1}^{n-1}(n-q)\binom{q+1}{2}$ f_{ij}^{kl}'s. From these, we exclude $n-1$ y-variables, namely, y_{1n}, $\ldots$, $y_{(n-1)n}$, and $\sum_{q=1}^{n-1}(n-q)q$ f-variables, namely, f_{kl}^{kl}, $f_{(k+1)l}^{kl}$, $\ldots$ $f_{(l-1)l}^{kl}$ for all (k, l)'s with $1 \leq k < l \leq n$. Then the number of excluded variables are the same

as the rank of the equality constraints. In addition we exclude y_{ab} that the number of remaining variables is $(n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24} - 1$. Hereafter, denote by E and R the sets of excluded and remaining variables, respectively. I.e.

$$R = \{y_{ij} : ij \in A,\ j \neq n\} \cup \{x_{kl} : kl \in A\} \cup \{f_{ij}^{kl} : 1 \leq k \leq i < j \leq l \leq n,\ j \neq l\}. \tag{2.16}$$

Lower-Triangularization Technique

We will find the vectors in S which, if their coordinates are restricted to the remaining variables R and stacked in rows, constitute a lower triangular matrix with nonzero diagonal elements. Since $|R| = (n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24} - 1$, the proof will then be completed. To do so, in each of the cases in the subsequent proof, we extend the current lower triangular matrix by augmenting the same number of columns and rows, which correspond to a new set of variables from R and the subvectors of new independent vectors in S.

	current variables			new variables		
current	$+$					
indep	$\times$	$\ddots$				
vectors	$\times$	$\dots$	$+$			
new	$\times$	$\times$	$\times$	$+$		
indep	$\times$	$\times$	$\times$		$+$	
vectors	$\times$	$\times$	$\times$			$+$

Figure 2.3: Lower triagularization technique

New subvector has unique nonzero entries on distinct coordinates of new variables so that they, permutated if necessary, extend the current matrix with nonzero diagonal elements as indicated in Figure 2.3. (The empty space means 0's.) To sustain the lower triangularity we need to carefully order the introduced variables so that

existing subvectors have no nonzero entries on the new coordinates of the variables introduced subsequently. This property will be referred to as *Invariant Property*. The procedure is repeated until every variable of R appears in a column of the matrix. The subsequent part of the proof is therefore about how to find independent vectors from S and why the order of variables adopted in the proof maintains Invariant Property throughout. For an easy reference, we summarize, in advance, which set of variables will be added in columns in which (sub)case of the subsequent proof.

<u>y-variables</u>

Remaining y-variables are $\{y_{ij} : ij \in A \setminus ab,\ j \neq n\} = \{y_{ij} : ij \in A,\ j \neq n\ ij \notin A_{23}\}$ (**Case 1-1**) $\cup \{y_{ij} : ij \in A_{23} \setminus ab\}$ (**Case 2-5**)

<u>x-variables</u>

Recall no x-variable was excluded. $\{x_{kl} : 1 \leq k < l \leq n\} = \{x_{kl} : 1 \leq k < l \leq n,\ (k,l) \notin V_2 \times V_3\}$ (**Case 1-2-1**) $\cup \{x_{kl} : 1 \leq k < l \leq n,\ k < l,\ (k,l) \in V_2 \times V_3\}$ (**Case 2-1**).

<u>f-variables</u>

The f-variables in R are $\{f_{ij}^{kl} : 1 \leq k \leq i < j \leq l \leq n,\ j \neq l\} = \{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, jl \notin A_{23}\}$ (**Case 1-2-2**, **1-2-3**) $\cup \{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, \text{or } jl \in A_{23}\}$ (**Remaining cases**).

We summarize in Table 2.1 the variable sets of $R \setminus \{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, \text{or } jl \in A_{23}\}$ for extending M and their corresponding cases in the subsequent proof.

In comparison with (2.16), we see that the sets of Table 2.1 exhaust the y and x-variables, and f-variables of R except the variables in $\{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, \text{or } jl \in A_{23}\}$.

Table 2.1: The partition of the y, x-variables, and the f-variables $\{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, jl \notin A_{23}\}$ in R for extension of M and the corresponding cases.

Row No.	Variables	Conditions	Sets	Cases
1	y_{ij}	$ij \in A,\ j < n,\ ij \notin A_{23}$	$A_1 \cup A_2$	**1-1**
2	y_{ij}	$ij \in A,\ j < n,\ ij \in A_{23},\ ij \neq ab$	A_3	**2-5**
3	x_{kl}	$1 \leq k < l \leq n,\ (k,l) \notin V_2 \times V_3$	B_1	**1-2-1**
4	x_{kl}	$1 \leq k < l \leq n,\ (k,l) \in V_2 \times V_3$	B_2	**2-1**
5	f_{ij}^{kl}	$1 \leq k = i < j < l \leq n,\ ki, ij, jl \notin A_{23}$	Q_1	**1-2-2**
6	f_{ij}^{kl}	$1 \leq k < i < j < l \leq n,\ ki, ij, jl \notin A_{23}$	Q_2	**1-2-3**

The remaining f-variables $\{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, \text{or } jl \in A_{23}\}$ of R are enumerated in Table 2.2 according to which of V_q, $1 \leq q \leq 4$ each of k, i, j, and l belongs to. Since at least one arc is used from V_2 to V_3, we have $H(4,4) - 2H(3,4) + H(2,4) = 10$ combinations. (Here $H(n,r)$ is the number of possible ways for choosing r from n objects when repetition allowed.) Each combination is subdivided further into cases to additional conditions on k, i, j, and/or l that are relevant to maintaining Invariant Property. In the last column, indicated are the cases of the proof in which the corresponding variables appear in columns and their values as diagonal elements of the matrix. In the table p stands for the node in V_1 such that $\pi_2(pa) = b$ in the assumption 2) of the theorem.

It is worth to emphasize that all the variables in R are exhaustively partitioned in Table 2.1 and 2.2, or 23 (Sub)cases in the subsequent proof.

We will construct independent vectors in S by taking the difference of two vectors on F. More specifically, we incrementally find sets of affinely independent vectors w^1, w^2, ..., w^K on F whose translation into S, $w^2 - w^1$, ..., $w^K - w^1$, if restricted to the subset of variables discussed above, extend a lower triangular matrix in which

Table 2.2: The partition of the remaining f-variables $\{f_{ij}^{kl} : 1 \leq k \leq i < j < l \leq n,\ ki, ij, \text{or } jl \in A_{23}\}$ in R for incremental extension of M and the corresponding cases.

Row No.	V_1	V_2	V_3	V_4	Conditions	Sets	Cases
7	$k \leq i$	j	l		$(k,l) \neq \pi(ij), \pi_2(ij) = b$	Q_3	**2-2**
8					$(k,l) \neq \pi(ij), \pi_2(ij) \neq b$	Q_9	**2-9**
9					$(k,l) = \pi(ij)$	Q_{10}	**2-10**
10	k	$i < j$	l		$(k,l) \neq (p,b), a = i$	Q_4	**2-3**
11					$(k,l) \neq (p,b), a \neq i$	Q_6	**2-6**
12					$(k,l) = (p,b)$	Q_7	**2-7**
13	k	i	$j < l$		$(k,l) \neq (p,b)$	Q_5	**2-4**
14					$(k,l) = (p,b)$	Q_8	**2-8**
15	k	i	j	l		Q_{19}	**2-19**
16		$k \leq i < j$	l			Q_{16}	**2-16**
17		$k \leq i$	$j < l$			Q_{17}	**2-17**
18		$k \leq i$	j	l	$j = b$	Q_{11}	**2-11**
19					$j \neq b$	Q_{13}	**2-13**
20		k	$i < j < l$			Q_{18}	**2-18**
21		k	$i < j$	l	$j = b$	Q_{12}	**2-12**
22					$j \neq b$	Q_{14}	**2-14**
23		k	i	$j < l$		Q_{15}	**2-15**

every diagonal element is nonzero. The subsequent part of proof consists of two main cases, Case 1 and 2, depending on whether the vectors w^j's satisfy the equation in (2.15) with 0's or 1's on both sides.

Case 1: '0=0'

In Case 1, we use feasible solutions w^j on F satisfying the equation of (2.15) with 0's on both sides, which is the case when path does not have an arc from A_{23}: $y_{ij} = 0$ for every $ij \in A_{23}$. Throughout the case, we reserve w^0 to denote the feasible solution with a single nonzero entry, $y_{1n} = 1$.

Case 1-1 (Row No. 1) Case 1-1 begins with construction of a lower triangular

matrix M with the aforementioned properties with the coordinates $\{y_{ij} : ij \in A, j < n, ij \notin A_{23}\}$. In doing so, we perform a two-step extension to illustrate the principle of Lower Triangularization Technique.

Consider the feasible solution w that uses the path $1 - j - n$ for each $1 < j < n$ but has $x = 0$ and $f = 0$. Then the nonzero variables in w are y_{1j} and y_{jn}. Define $A_1 = \{1j \in A : 1 < j < n\}$ endowed with an arbitrary but fixed order on the elements. Then $\{y_{ij} : ij \in A_1\} \subseteq R$. The subvector y_{A_1} obtained from w by restricting its coordinates to A_1 has a single nonzero entry, namely $y_{1j} = 1$. Construct a square matrix M by stacking as rows the subvectors y_{A_1} so that $1j$'s with $y_{1j} = 1$ are in the same order as in A_1. Then M is clearly a lower triangular matrix with a nonzero diagonal. Since $y_{jn} \in E$ (recall, the set of excluded variables) for $1 < j < n$, the rows will not have any zero entry when they are extended by any set of coordinates from R. Thus M has Invariant Property.

To extend M, consider the feasible solution w using Boobosang path $1-i-j-n$ for each $ij \in A$, $1 < i < j < n$, and $ij \notin A_{23}$, but having $x = 0$ and $f = 0$. The nonzero entries of w are $y_{1i} = y_{ij} = y_{jn} = 1$. Let $A_2 = \{ij \in A : 1 < i < j < n,\ ij \notin A_{23}\}$. Note that the coordinates of y_{A_1} and y_{A_2} exhaust the y_{ij}'s in R with $ij \notin A_{23}$. Extend the columns of M with the coordinates corresponding to the variables, y_{ij}, $ij \in A_2$. Since the variables are in R and M has Invariant Property, the new columns contains no nonzero entry. Thus if we append to M in rows the subvectors (y_{A_1}, y_{A_2}) of w's so that ij's with $y_{ij} = 1$ are in the same order as in A_2, the obtained matrix is again a lower triangular matrix with a nonzero diagonal. Also note that in each feasible solution w, the nonzero variables beside y_{ij}, $y_{1i} \in y_{A_1}$, and $y_{jn} \in E$. That is, Invariant Property is maintained in M.

Note w^0 has no nonzero entry on R since the only nonzero variable $y_{1n} \in E$. Therefore w and $w - w^0$ if their coordinates are restricted to R. As the columns of M will be extended up to R, we can consider the rows of M as the subvectors of $w - w^0 \in S$.

Case 1-2 Consider the feasible solution w using the path $1 - k - i - j - l - n$ for each quadruple $1 \leq k \leq i < j \leq l \leq n$ such that ki, ij, $jl \notin A_{23}$ (and thus satisfying the equation of (2.15) with 0's on both sides). It has a single trade $x_{kl} = \varepsilon$, where $var\varepsilon = \min\{U - d(\pi(ij)), d(k,l) : ij \in A_{12}, kl \in A\} > 0$.

Then the nonzero variables are $y_{1k} = y_{ki} = y_{ij} = y_{jl} = y_{ln} = 1$, $x_{kl} = \varepsilon$, and $f_{ki}^{kl} = f_{ij}^{kl} = f_{jl}^{kl} = \varepsilon$. Case 1-2 will be divided further into three subsubcases depending on whether $k = i$ and/or $j = l$ as in Figure 2.4

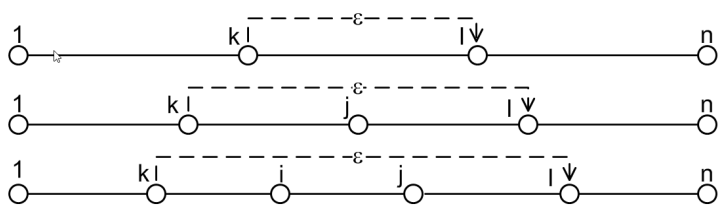

Figure 2.4: The three subcases of Case 1-2

Case 1-2-1 (Row No. 3) Suppose in addition $k = i$ and $j = l$. Then $x_{kl} = f_{kl}^{kl} = \varepsilon$, $y_{1k} = y_{kl} = y_{ln} = 1$, and the other variables are all 0. Now we extend the current matrix M with the columns corresponding to x_{B_1} where $B_1 = \{(k,l) : 1 \leq k < l \leq n,\ (k,l) \notin V_2 \times V_3\}$. Then $x_{B_1} \subseteq R$ (Note we are abusing notation for short by not distinguishing a vector and the set of its entries.) due to Invariant Property of M, the new columns should be all zero vectors.

We append the set of subvectors $(y_{A_1 \cup A_2}, x_{B_1})$ of $w - w^0$'s to M in rows so that

(k,l) with $x_{kl} = \varepsilon$ are in the same order as (k,l)'s in B_1. Then the extended M is also lower triangular with nonzero diagonal. Since $f_{kl}^{kl} \in E$, $y_{1k} \in y_{A_1}$, $y_{kl} \in y_{A_2}$, and $y_{ln} \in E$, Invariant Property of M is maintained.

Case 1-2-2 (Row No. 5): Suppose $k = i$ and $j < l$. Then $y_{1k} = y_{kj} = y_{jl} = y_{ln} = 1$, $x_{kl} = \varepsilon$, $f_{kj}^{kl} = f_{jl}^{kl} = \varepsilon$, and the other variables are all 0 in w. Consider the subvectors $(y_{A_1 \cup A_2}, x_{B_1}, f_{Q_1})$ of $(w - w^0)$'s where $Q_1 = \{(k,i,j,l) : 1 \leq k = i < j < l \leq n,\ ij, jl \notin A_{23}\}$. Then $Q_1 \subseteq R$. Extend M with columns corresponding to f_{Q_1} which should be all zero vectors due to Invariant Property of M. Append the subvectors in rows to M so that (k,k,j,l) with $f_{kj}^{kl} = 1$ are in the same order as (k,k,j,l)'s in Q_1. Then M is a lower triangular matrix with a nonzero diagonal.

Invariant property again holds for the extended M since the nonzero variables of each w belongs to the existing coordinates: $y_{1k} \in y_{A_1}$, y_{kj}, $y_{jl} \in y_{A_2}$, $y_{ln} \in E$, $x_{kl} \in x_{B_1}$, and $f_{jl}^{kl} \in E$.

Case 1-2-3 (Row No. 6): Suppose $k < i$ and $j < l$. Then, the nonzero entries of each w are $y_{1k} = y_{ki} = y_{ij} = y_{jl} = y_{ln} = 1$, $x_{kl} = \varepsilon$, and $f_{ki}^{kl} = f_{ij}^{kl} = f_{jl}^{kl} = \varepsilon$. Consider the subvectors of $(y_{A_1 \cup A_2}, x_{B_1}, f_{Q_1 \cup Q_2})$ of $(w - w^0)$'s. Extend M with columns corresponding to f_{Q_2} where $Q_2 = \{(k,i,j,l) :\ 1 \leq k < i < j < l \leq n,\ ki, ij, jl \notin A_{23}\}$. Then $Q_2 \subseteq R$ and hence the new columns should all zero vectors due to Invariant Property of M. Thus if we append the subvectors to M so that (k,i,j,l) with $f_{ij}^{kl} = \varepsilon$ are in the same order as (k,i,j,l)'s in Q_2, we get a lower triangular matrix with a nonzero diagonal.

M inherits Invariant Property since the nonzero variables beside f_{Q_2} satisfy $y_{1k} \in y_{A_1}$, y_{ki}, y_{ij}, $y_{jl} \in y_{A_2}$, $y_{ln} \in E$, $x_{kl} \in x_{B_1}$, $f_{ki}^{kl} \in f_{Q_1}$, and $f_{jl}^{kl} \in E$.

Case 2: '1=1' In Case 2, we use feasible solutions $w \in F$ that satisfy (2.15) with 1's on both sides. More specifically, in each of the subsequent cases, we will find independent vectors $w^2 - w^1$ where w^1 and w^2 feasible solutions that carry a positive flow on an arc of A_{23}. To secure enough of them, the assumptions of the theorem on the existence of particular f-terms in the equation of (2.15) play crucial roles.

Case 2-1 (Row No. 4) From the assumption 2) of the theorem, on the left hand side of the equation of (2.15) is the term $f^{pb}_{(a-1)a}/d(p,b)$ for $p \in V_1$. For each $ij \in A_{23}$, consider the path 1-p-$(a-1)$-a-i-j-b-n where $a \leq i < j \leq b$. we consider two feasible solutions w^1 and w^2 a path where w^1 has a single trade, $x_{pb} = d(p,b)$ while w_2 has an additional trade $x_{ij} = \varepsilon$.

Both w_1 and w_2 satisfy (2.15) with 1's on both sides and hence $w^2 - w^1$ is a vector in S with the nonzero entries $x_{ij} = f^{ij}_{ij} = \varepsilon$, $(i,j) \in V_2 \times V_3$. Let $x_{B_2} := \{x_{kl} : (k,l) \in V_2 \times V_3\}$. Extend M with the columns corresponding to x_{B_2}. Then the new columns are all zero, due to Invariant Property of M. Append to M in rows the set of subvectors obtained by restricting $w^2 - w^1$ to $(y_{A_1 \cup A_2}, x_{B_1}, f_{Q_1 \cup Q_2}, x_{B_2})$ in order that (i,j)'s with $x_{ij} = \varepsilon$ are in the same order as (i,j)'s in B_2. The M remains to be lower triangular.

Since the remaining nonzero entry $f^{ij}_{ij} \in E$, Invariant Property is maintained.

Note 1 Note that all the x-variables have appeared in the columns of M. Thus in arguing Invariant Property hereafter we may well omit discussing x-variables.

Case 2-2 (Row No. 7) For each quadruple (k,i,j,l) such that $k,i \in V_1$, $k \leq i$, $j \in V_2$, $l \in V_3$, $(k,l) \neq \pi(ij)$, and $\pi_2(ij) = b$, consider the path $1 - \pi_1(ij) \leftrightarrow k - i - j - l - \pi_2(ij)(= b) - n$ where $u \leftrightarrow v$ means either $u \leq v$ or $v < u$. Let w^1

be the feasible solution with a single nonzero trade variable $x_{\pi(ij)} = d(\pi(ij))$ while w^2 with the additional nonzero trade variable $x_{kl} = \varepsilon$. Note that w_1 and w_2 satisfy (2.15) with 1's on both sides.

Then $w_2 - w_1 \in S$ has as the nonzero variables, $x_{kl} = f^{kl}_{ki} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$ if $\pi_1(ij) \leq k$; $x_{kl} = f^{kl}_{k\pi_1(ij)} = f^{kl}_{\pi_1(ij)i} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$ if $\pi_1(ij) > k$. Consider the set of the subvectors of $w_2 - w_1$ corresponding to the variables $(y_{A_1 \cup A_2}, x_{B_1}, f_{Q_1 \cup Q_2}, x_{B_2}, f_{Q_3})$, where $Q_3 = \{(k, i, j, l) : k \in V_1, i \in V_1, k \leq i, j \in V_2, l \in V_3, \pi_2(ij) = b, (k, l) \neq \pi(ij)\}$. Extend M with the columns corresponding to f_{Q_3}, all zero vectors due to Invariant Property of M. Append the subvectors to M in rows so that (k, i, j, l) with $f^{kl}_{ij} = \varepsilon$ are in the same order as in Q_3. Then the obtained matrix is a lower triangular matrix with a nonzero diagonal as desired. If $\pi_1(ij) \leq k$, the fact that $f^{kl}_{ki} \in Q_1$ and $f^{kl}_{jl} \in E$ maintains Invariant Property of M. If on the other hand $\pi_1(ij) > k$, that $f^{kl}_{k\pi_1(ij)} \in Q_1$ and $f^{kl}_{\pi_1(ij)i} \in Q_2$ guarantees Invariant Property.

Case 2-3 (Row No. 10) For brevity from now on we present the feasible solutions w^i by specifying only their paths P^i for $i = 1, 2$ and their trade variables x, the nonzero entries of $w^2 - w^1$, and the new coordinate set and their values to appear as new diagonal entries. Also we argue Invariant Property by specifying which of the coordinate sets so far or E each variable having a nonzero entry in $w^2 - w^1$ belongs to.

<u>Feasible solution w^1</u> $P_1 = 1\text{-}p \leftrightarrow k\text{-}(a-1)\text{-}i(=a)\text{-}j\text{-}l\text{-}b\text{-}n$, with $k \in V_1$ $j \in V_2$, $a < j$, $l \in V_3$, and $(k, l) \neq (p, b)$. $x_{pb} = d(p, b)$.

<u>Feasible solution w^2</u> $P_2 = P_1$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

<u>Nonzero entries of $w^2 - w^1$</u> $x_{kl} = f^{kl}_{k(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{aj} = f^{kl}_{jl} = \varepsilon$ if $p \leq k$. $x_{kl} = f^{kl}_{kp} = f^{kl}_{p(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{aj} = f^{kl}_{jl} = \varepsilon$ if $p > k$.

New diagonal entries $f^{kl}_{aj} = \varepsilon$ for $(k,a,j,l) \in Q_4 := \{(k,a,j,l) : k \in V_1, a < j, j \in V_2, l \in V_3, (k,l) \neq (p,b)\}$.

Invariant Property If $p \leq k$, $f^{kl}_{k(a-1)} \in f_{Q_1}$, $f^{kl}_{(a-1)a} \in f_{Q_3}$, and $f^{kl}_{jl} \in E$. If $p > k$, $f^{kl}_{kp} \in f_{Q_1}$, $f^{kl}_{p(a-1)} \in f_{Q_2}$, $f^{kl}_{(a-1)a} \in f_{Q_3}$, and $f^{kl}_{jl} \in E$.

Case 2-4 (Row No. 13) Feasible solution w^1 $P_1 = 1\text{-}p \leftrightarrow k\text{-}(a-1)\text{-}a\text{-}i\text{-}j\text{-}l\text{-}b\text{-}n$, $k \in V_1$, $ij \in A_{23}$, $j < l$, $l \in V_3$, and $(k,l) \neq (p,b)$. $x_{pb} = d(p,b)$.

Feasible solution w^2 $P_2 = P_1$. $x_{pb} = d(p,b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $x_{kl} = f^{kl}_{k(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{ai} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$ if $p \leq k$. $x_{kl} = f^{kl}_{kp} = f^{kl}_{p(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{ai} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$ if $p > k$.

New diagonal entries $f^{kl}_{ij} = \varepsilon$ for $(k,i,j,l) \in Q_5 := \{(k,i,j,l) : k \in V_1, ij \in A_{23}, j < l, (k,l) \neq (p,b)\}$.

Invariant Property If $p \leq k$, $f^{kl}_{k(a-1)} \in f_{Q_1}$, $f^{kl}_{(a-1)a} \in f_{Q_3}$, $f^{kl}_{ai} \in f_{Q_4}$, and $f^{kl}_{jl} \in E$. If $p > k$, $f^{kl}_{kp} \in f_{Q_1}$ and $f^{kl}_{p(a-1)} \in f_{Q_2}$, $f^{kl}_{(a-1)a} \in f_{Q_3}$, $f^{kl}_{ai} \in f_{Q_4}$, and $f^{kl}_{jl} \in E$.

Case 2-5 (Row No. 2) Feasible solution w^1 $P_1 = 1\text{-}n$. $x = 0$.

Feasible solution w^2 $P_2 = 1\text{-}q\text{-}r\text{-}a\text{-}i\text{-}j\text{-}b\text{-}n$, $i \in V_2$, $j \in V_3$, $(i,j) \neq (a,b)$. $x_{qb} = d(q,b)$.

Nonzero entries of $w^2 - w^1$ $x_{qb} = f^{qb}_{qr} = f^{qb}_{ra} = f^{qb}_{ai} = f^{qb}_{ij} = f^{qb}_{jb} = d(q,b)$, $y_{1q} = y_{qr} = y_{ra} = y_{ai} = y_{ij} = y_{jb} = y_{bn} = 1$, $y_{1n} = -1$.

New diagonal entries $y_{ij} = 1$ for $ij \in A_3 = \{ij \in A :, j < n, ij \in A_{23}, \text{ and } ij \neq ab\}$.

Invariant Property

$f^{qb}_{qr} \in f_{Q_1}$, $f^{qb}_{ra} \in f_{Q_3}$, $f^{qb}_{ai} \in Q_4$, $f^{qb}_{ij} \in f_{Q_5}$, $f^{qb}_{jb} \in E$, $y_{1q} \in y_{A_1}$, y_{qr}, y_{ra}, y_{ai}, $y_{jb} \in y_{A_2}$, and y_{bn}, $y_{1n} \in E$.

Note 2 All the x and y-variables in R have appeared in the columns of M. Hereafter, we may omit x and y-variables in arguing Invariable Property.

Case 2-6 (Row No. 11) <u>Feasible solution w^1</u> $P_1 = 1\text{-}p \leftrightarrow k\text{-}(a-1)\text{-}a\text{-}i\text{-}j\text{-}l\text{-}b\text{-}n$, $k \in V_1$, i, $j \in V_2$, $a < i < j$, $l \in V_3$, and $(k,l) \neq (p,b)$. $x_{pb} = d(p,b)$.

<u>Feasible solution w^2</u> $P_2 = P_1$. $x_{pb} = d(p,b)$ and $x_{kl} = \varepsilon$.

<u>Nonzero entries of $w^2 - w^1$</u> $f^{kl}_{k(a-1)}(f^{kl}_{kp}$ and $f^{kl}_{p(a-1)}$ if $k < p) = f^{kl}_{(a-1)a} = f^{kl}_{ai} = f^{kl}_{ij}$ $= f^{kl}_{jl} = \varepsilon$ if $p \leq k$.

<u>New diagonal entries</u> $f^{kl}_{ij} = \varepsilon$ for $(k,i,j,l) \in Q_6 := \{(k,i,j,l) : k \in V_1, i,j \in V_2, a < i < j, l \in V_3, (k,l) \neq (p,b)\}$.

<u>Invariant Property</u> $f^{kl}_{k(a-1)} \in f_{Q_1}(f^{kl}_{kp} \in f_{Q_1}$ and $f^{kl}_{p(a-1)} \in f_{Q_2}$ if $k < p)$, $f^{kl}_{(a-1)a} \in f_{Q_3}$, $f^{kl}_{ai} \in f_{Q_4}$, and $f^{kl}_{jl} \in E$.

Case 2-7 (Row No. 12) <u>Feasible solution w^1</u> $P_1 = 1\text{-}k(=p)\text{-}(a-1)\text{-}a\text{-}i\text{-}l(=b)\text{-}n$, i, $j \in V_2$, $i < j$. $x_{pb} = d(p,b)$.

<u>Feasible solution w^2</u> $P_2 = 1\text{-}k(=p)\text{-}(a-1)\text{-}a\text{-}i\text{-}j\text{-}l(=b)\text{-}n$, i, $j \in V_2$, $i < j$. $x_{pb} = d(p,b)$.

<u>Nonzero entries of $w^2 - w^1$</u> $f^{pb}_{ij} = f^{pb}_{jb} = d(p,b)$, $f^{pb}_{ib} = -d(p,b)$

<u>New diagonal entries</u> $f^{pb}_{ij} = d(p,b)$ for $(p,i,j,b) \in Q_7 := \{(p,i,j,b) : i,j \in V_2, i < j\}$.

<u>Invariant Property</u> f^{pb}_{jb}, and $f^{pb}_{ib} \in E$.

Case 2-8 (Row No. 14)

<u>Feasible solution w^1</u> $P_1 = 1\text{-}k(=p)\text{-}a-1\text{-}a\text{-}i\text{-}l(=b)\text{-}n$, $i \in V_2$, $j \in V_3$, and $j < b$. $x_{pb} = d(p,b)$.

<u>Feasible solution w^2</u> $P_2 = 1\text{-}k(=p)\text{-}a-1\text{-}a\text{-}i\text{-}j\text{-}l(=b)\text{-}n$, $i \in V_2$, $j \in V_3$, and $j < b$. $x_{pb} = d(p,b)$.

<u>Nonzero entries of $w^2 - w^1$</u> $f^{pb}_{ij} = f^{pb}_{jb} = d(p,b)$, $f^{pb}_{ib} = -d(p,b)$.

<u>New diagonal entries</u> $f^{pb}_{ij} = d(p,b)$ for $(p,i,j,b) \in Q_8 := \{(p,i,j,b) : i \in V_2, j \in V_3, j < b\}$.

Invariant Property f^{pb}_{jb}, and $f^{pb}_{ib} \in E$.

Case 2-9 (Row No. 8) Feasible solution w^1 $P_1 = 1\text{-}\pi_1(ij) \leftrightarrow k\text{-}i\text{-}j\text{-}l \leftrightarrow \pi_2(ij)\text{-}n$, $k, i \in V_1$, $k \leq i$, $j \in V_2$, $l \in V_3$, $\pi_2(ij) \neq b$ and $(k,l) \neq \pi(ij)$. $x_{\pi(ij)} = d(\pi(ij))$.

Feasible solution w^2 $P_2 = P_1$. $x_{\pi(ij)} = d(\pi(ij))$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ If $\pi_1(ij) \leq k$, then $f^{kl}_{ki} = \varepsilon$, or else, namely $\pi_1(ij) > k$, then $f^{kl}_{k\pi_1(ij)} = f^{kl}_{\pi_1(ij)i}\ ) = \varepsilon$. And $f^{kl}_{ij} = \varepsilon$. If $l \leq \pi_2(ij)$, then $f^{kl}_{jl} = \varepsilon$, or else, i.e. $l > \pi_2(ij)$, then $f^{kl}_{j\pi_2(ij)} = f^{kl}_{\pi_2(ij)l} = \varepsilon$.

New diagonal entries $f^{kl}_{ij} = \varepsilon$ for $(k,i,j,l) \in Q_9 := \{(k,i,j,l) : k, i \in V_1, k \leq i, j \in V_2, l \in V_3, \pi_2(ij) \neq b, (k,l) \neq \pi(ij)\}$.

Invariant Property If $\pi_1(ij) \leq k$, then $f^{kl}_{ki} \in f_{Q_1}$. If else, i.e. $\pi_1(ij) > k$ $f^{kl}_{k\pi_1(ij)} \in f_{Q_1}$ and $f^{kl}_{\pi_1(ij)i} \in f_{Q_2}$. If $l \leq \pi_2(ij)$, then $f^{kl}_{jl} \in E$. If, on the other hand, $l > \pi_2(ij)$ then $k \in V_1$, $j \in V_2$, $\pi_2(ij)$, $l \in V_3$. Then depending on $(k,l) = (p,b)$ or not, we have $f^{kl}_{j\pi_2(ij)} \in f_{Q_5}$ or f_{Q_8}.

Case 2-10 (Row No. 9) Feasible solution w^1 $P_1 = 1\text{-}n$, $i \in V_1$, $j \in V_2$, and $(k,l) = \pi(ij)$. $x = 0$.

Feasible solution w^2 $P_2 = 1\text{-}k(= \pi_1(ij))\text{-}i\text{-}j\text{-}l(= \pi_2(ij))\text{-}n$, $i \in V_1$, $j \in V_2$, and $(k,l) = \pi(ij)$. $x_{\pi(ij)} = d(\pi(ij))$.

Nonzero entries of $w^2 - w^1$ $f^{\pi(ij)}_{\pi_1(ij)i} = f^{\pi(ij)}_{ij} = f^{\pi(ij)}_{j\pi_2(ij)} = d(\pi(ij))$.

New diagonal entries $f^{\pi(ij)}_{ij} = d(\pi(ij))$ for $(\pi_1(ij), i, j, \pi_2(ij)) \in Q_{10} := \{(\pi_1(ij), i, j, \pi_2(ij)) : i \in V_1, j \in V_2\}$.

Invariant Property $f^{\pi(ij)}_{\pi_1(ij)i} \in f_{Q_1}$, and $f^{\pi(ij)}_{\pi_1(ij)i} \in E$.

Case 2-11 (Row No. 18) Feasible solution w^1 $P_1 = 1\text{-}p\text{-}(a-1)\text{-}a\text{-}k\text{-}i\text{-}j(= b)\text{-}l\text{-}n$, $k, i \in V_2$, $k \leq i$, and $l \in V_4$. $x_{pb} = d(p,b)$.

Feasible solution w^2 $P_2 = P_1$. $x_{pb} = d(p,b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{ki}^{kl} = f_{ib}^{kl} = f_{bl}^{kl} = \varepsilon$.

New diagonal entries $f_{ib}^{kl} = \varepsilon$ for $(k,i,b,l) \in Q_{11} := \{(k,i,b,l) : k,i \in V_2, k \leq i, l \in V_4\}$.

Invariant Property $f_{ki}^{kl} \in f_{Q_1}$, and $f_{bl}^{kl} \in E$.

Case 2-12 (Row No. 21) Feasible solution w^1 $P_1 = 1$-s-t-a-k-c-i-l-n, $k \in V_2$, $i \in V_3$, $i < b$ and $l \in V_4$. $x_{sb} = d(s,b)$ and $x_{kl} = \varepsilon$.

Feasible solution $w^2 P_2 = 1$-s-t-a-k-c-i-b-l-n, $k \in V_2$, $i \in V_3$, and $l \in V_4$. $x_{sb} = d(s,b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{ib}^{kl} = f_{bl}^{kl} = \varepsilon$, and $f_{il}^{kl} = -\varepsilon$.

New diagonal entries $f_{ib}^{kl} = \varepsilon$ for $(k,i,b,l) \in Q_{12} := \{(k,i,b,l) : k \in V_2, i \in V_3, i < b, l \in V_4\}$.

Invariant Property f_{bl}^{kl}, $f_{il}^{kl} \in E$.

Case 2-13 (Row No. 19) Feasible solution w^1 $P_1 = 1$-p-$(a-1)$-a-k-i-b-l-n, $k,i \in V_2$, $k \leq i$, $j \in V_3$, $j < b$ and $l \in V_4$. $x_{pb} = d(p,b)$ and $x_{kl} = \varepsilon$.

Feasible solution w^2 $P_2 = 1$-p-$(a-1)$-a-k-i-j-b-l-n, $k,i \in V_2$, $k \leq i$, $j \in V_3$, $j < b$ and $l \in V_4$. $x_{pb} = d(p,b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{ij}^{pb} = f_{jb}^{pb} = d(p,b)$, $f_{ib}^{pb} = -d(p,b)$, $f_{ij}^{kl} = f_{jb}^{kl} = \varepsilon$, and $f_{ib}^{kl} = -\varepsilon$.

New diagonal entries $f_{ij}^{kl} = \varepsilon$ for $(k,i,j,l) \in Q_{13} := \{(k,i,j,l) : k,i \in V_2, k \leq i, j \in V_3, j < b, l \in V_4\}$.

Invariant Property $f_{ij}^{pb} \in f_{Q_8}$, f_{jb}^{pb}, $f_{ib}^{pb} \in E$, $f_{jb}^{kl} \in f_{Q_{12}}$, and $f_{ib}^{kl} \in f_{Q_{11}}$.

Case 2-14 (Row No. 22) Feasible solution w^1 $P_1 = 1$-p-$(a-1)$-a-k-i-j-b-l-n, $k \in V_2$, $i,j \in V_3$, $i < j < b$, and $l \in V_4$. $x_{pb} = d(p,b)$.

Feasible solution w^2 $P_2 = P_1$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{ki}^{kl} = f_{ij}^{kl} = f_{jb}^{kl} = f_{bl}^{kl} = \varepsilon$.

New diagonal entries $f_{ij}^{kl} = \varepsilon$ for $(k, i, j, l) \in Q_{14} := \{(k, i, j, l) : k \in V_2, i, j \in V_3, i < j < b, l \in V_4\}$.

Invariant Property $f_{ki}^{kl} \in f_{Q_{13}}$, $f_{jb}^{kl} \in f_{Q_{12}}$, and $f_{bl}^{kl} \in E$.

Case 2-15 (Row No. 23) Feasible solution w^1 $P_1 = 1$-s-t-a-k-c-i-j-l-n, $k \in V_2$, $i \in V_3$, $j, l \in V_4$, and $j < l$. $x_{sc} = d(s, c)$.

Feasible solution w^2 $P_2 = P_1$. $x_{sc} = d(s, c)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{kc}^{kl} = f_{ci}^{kl} = f_{ij}^{kl} = f_{jl}^{kl}$ ε.

New diagonal entries $f_{ij}^{kl} = \varepsilon$ for $(k, i, j, l) \in Q_{15} := \{(k, i, j, l) : k \in V_2, i \in V_2, j, l \in V_3, j < l\}$.

Invariant Property $f_{kc}^{kl} \in f_{Q_{13}}$, $f_{ci}^{kl} \in f_{Q_{14}}$, and $f_{jl}^{kl} \in E$.

Case 2-16 (Row No. 16) Feasible solution w^1 $P_1 = 1$-p-$(a-1)$-a-k-i-l-b-n, $k, i, j \in V_2$, $k \leq i < j$, and $l \in V_3$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

Feasible solution w^2 $P_2 = 1$-p-$(a-1)$-a-k-i-j-l-b-n, $k, i, j \in V_2$, $k \leq i < j$, and $l \in V_3$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

Nonzero entries of $w^2 - w^1$ $f_{ij}^{pb} = f_{jl}^{pb} = d(p, b)$, $f_{il}^{pb} = -d(p, b)$, $f_{ij}^{kl} = f_{jl}^{kl} = \varepsilon$, $f_{il}^{kl} = -\varepsilon$.

New diagonal entries $f_{ij}^{kl} = \varepsilon$ for $(k, i, j, l) \in Q_{16} := \{(k, i, j, l) : k, i, j \in V_2, k \leq i < j, l \in V_3\}$.

Invariant Property

$f_{ij}^{pb} \in f_{Q_7}$, f_{jl}^{pb}, $f_{il}^{pb} \in f_{Q_8}$, and f_{jl}^{kl}, $f_{il}^{kl} \in E$.

Case 2-17 (Row No. 17) Feasible solution w^1 $P_1 = 1$-p-$(a-1)$-a-k-i-j-l-b-n, $k, i \in$

V_2, $k \leq i$, $j, l \in V_3$, and $j < l$. $x_{pb} = d(p, b)$.

<u>Feasible solution w^2</u> $P_2 = P_1$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

<u>Nonzero entries of $w^2 - w^1$</u> $f^{kl}_{ki)} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$.

<u>New diagonal entries</u> $f^{kl}_{ij} = \varepsilon$ for $(k, i, j, l) \in Q_{17} := \{(k, i, j, l) : k, i \in V_2, k \leq i, j, l \in V_3, j < l\}$.

<u>Invariant Property</u> $f^{kl}_{ki} \in f_{Q_{16}}$ and $f^{kl}_{jl} \in E$.

Case 2-18 (Row No. 20) <u>Feasible solution w^1</u> $P_1 = 1$-p-$(a-1)$-a-k-i-j-l-b-n, $k \in V_2$, $i, j, l \in V_3$, and $i < j < l$. $x_{pb} = d(p, b)$.

<u>Feasible solution w^2</u> $P_2 = P_1$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

<u>Nonzero entries of $w^2 - w^1$</u> $f^{kl}_{ki)} = f^{kl}_{ij} = f^{kl}_{jl} = \varepsilon$.

<u>New diagonal entries</u> $f^{kl}_{ij} = \varepsilon$ for $(k, i, j, l) \in Q_{18} := \{(k, i, j, l) : k \in V_2, i, j, l \in V_3, i < j < l\}$.

<u>Invariant Property</u> $f^{kl}_{ki} \in f_{Q_{17}}$ and $f^{kl}_{jl} \in E$.

Case 2-19 (Row No. 15) <u>Feasible solution w^1</u> $P_1 = 1$-$p \leftrightarrow k$-$a-1$-a-i-j-b-l-n, $k \in V_1$, $i \in V_2$, $j \in V_3$, and $l \in V_4$. $x_{pb} = d(p, b)$.

<u>Feasible solution w^2</u> $P_2 = P_1$. $x_{pb} = d(p, b)$ and $x_{kl} = \varepsilon$.

<u>Nonzero entries of $w^2 - w^1$</u> If $p \leq k$, then $f^{kl}_{k(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{ai} = f^{kl}_{ij} = f^{kl}_{jb} = f^{kl}_{bl} = \varepsilon$. If $p > k$, then $f^{kl}_{kp} = f^{kl}_{p(a-1)} = f^{kl}_{(a-1)a} = f^{kl}_{ai} = f^{kl}_{ij} = f^{kl}_{jb} = f^{kl}_{bl} = \varepsilon$.

<u>New diagonal entries</u> $f^{kl}_{ij} = \varepsilon$ for $(k, i, j, l) \in Q_{19} := \{(k, i, j, l) : k \in V_1, i \in V_2, j \in V_3, l \in V_4\}$.

<u>Invariant Property</u> If $p \leq k$, then $f^{kl}_{k(a-1)} \in f_{Q_1}$, $f^{kl}_{(a-1)a}$, f^{kl}_{ai}, $f^{kl}_{jb} \in f_{Q_2}$, and $f^{kl}_{bl} \in E$. If $p > k$, then $f^{kl}_{kp} \in f_{Q_1}$, $f^{kl}_{p(a-1)} \in f_{Q_2}$, $f^{kl}_{(a-1)a}$, f^{kl}_{ai}, $f^{kl}_{jb} \in f_{Q_2}$, and $f^{kl}_{bl} \in E$.

As claimed in Table 2.2, the above cases have exhaust the variables of R as the columns of the lower triangular matrix M whose diagonal elements are all nonzero.

Each row of M is a vector in S a subspace parallel with F. Therefore, the dimension of F is $|R| = (n-1)^2 + \frac{(n-2)(n-1)n(n+1)}{24} - 1$. Hence by Proposition 2.2.2 and the observation that $\dim F < \dim P_{int}$, F is a facet. $\square$

Separation for 3-Criteria Inequality

Now we propose an separation algorithm for 3-Criteria Inequality.

Algorithm 2.3.3. ***Separation algorithm for 3-Criteria Inequality***

1. *For each* $1 \leq i < c \leq b < n$, *compute* $y_{i[c,b]} := \sum_{j=[c,b]} y_{ij}$. *Since* $y_{i[c,b-1]} + y_{ib}$ *has the same value, each can be calculated in* $O(1)$ *time, so the overall computation time is* $O(n^3)$.

2. *For each* $1 < a < c \leq b < n$, *compute* $y_{[a,c,b]} := \sum_{i \in [a,c-1]} y_{i[c,b]}$. *This can be calculated with an operation* $y_{[a-1,c,b]} + y_{a[c,b]}$, *so the overall computation time is* $O(n^3)$.

3. *For each* $1 \leq i < j \leq l < n$, *compute* $f_{ij}^{*l} := \max_{k \leq i} f_{ij}^{kl}$. *Each can be obtained by comparing* $O(n)$ *values, so the overall computation time is* $O(n^4)$.

4. *For each* $1 \leq i < j \leq c \leq b < n$, *compute* $f_{ij}^{*[c,b]} := \max_{l \in [c,b]} f_{ij}^{*l}$. *This can be calculated with an operation by* $\max\{f_{ij}^{*b}, f_{ij}^{[c,b-1]}\}$, *so the overall computation time is* $O(n^4)$.

5. *For each* $1 \leq i < j \leq c \leq b < n$, *compute* $f_i^{[a,c,b]} := \sum_{j \in [a,c-1]} f_{ij}^{*[c,b]}$. *This can be calculated with an operation by* $f_i^{[a+1,c,b]} + f_{ia}^{*[c,b]}$, *so the overall computation time is* $O(n^4)$.

6. *For each a, compute $f^{[a,c,b]} := \sum_{i \in [1,a-1]} f_i^{[a,c,b]}$. Each takes $O(n)$ time, so the overall computation time is $O(n^4)$.*

7. *Finally, for each $1 \leq a \leq c < b < n$, compute the maximum value of $f^{[a,c,b]} - y_{[a,c,b]}$, which represents the maximum violation of the 3-Criteria Inequality. This can be computed in $O(n^3)$ time.*

The above algorithm is a separation algorithm for 3-Criteria Inequality with a computation time of $O(n^4)$.

Proposition 2.3.4. *There exists an $\theta(n^4)$-time separation algorithm for 3-Criteria Inequality.*

Since reading the value of flow variable f takes $O(n^4)$ time, we can see that this separation algorithm provides the best result modulo constant.

2.3.2 Other Valid Inequalities

In this subsection, we present several families of valid inequalities other than the 3-Criteria inequality.

Valid Inequalities for uncapacitated Case

First, consider the uncapacitated sMTP when there is no capacity constraint. Define $\bar{f}_{ij}^{kl} := f_{ij}^{kl}/d(k,l)$ as the normalized f_{ij}^{kl}, and express the constraint in terms of $\bar{f}$ and y. In this case of uncapacitated sMTP, there are no parameters other than n. We will use $\bar{f}$ as a variable instead of f.

Definition 2.3.5. Let S be an arbitrary set of nodes and arcs. We define $\mathcal{P}(S)$ as the set of all 1-n paths that contain all elements of S. Depending on the situation,

we can also directly input the elements of S. For example, $\mathcal{P}(\{a_1, \cdots, a_k\})$ is simply denoted as $\mathcal{P}(a_1, \cdots, a_k)$. Let $\bar{f}_{ij}^{kl}$ (or (k, i, j, l)) **generate** $\mathcal{P}(k, l, ij)$ and y_{ij} (or (i, j)) **generate** $\mathcal{P}(ij)$. For any set of paths S that is included in $\mathcal{P}(k, l, ij)$ (or $\mathcal{P}(ij)$), we say that $\bar{f}_{ij}^{kl}$ (or y_{ij}, respectively) **covers** S.

Let the set of paths generated by the weighted sum of variables be the union of the sets generated by each path, allowing multiset. Restate the condition for the uncapacitated case of Theorem 2.2.1 using the terms "generate" and "cover": for $\sum a_{ij}^{kl} \bar{f}_{ij}^{kl} \leq \sum b_{ij} y_{ij}$ ($a_{ij}^{kl} \in \mathbb{Z}$, $b_{ij} \in \mathbb{Z}_0^+$) to be facet-defining inequality, the set generated by $a_{ij}^{kl} \bar{f}_{ij}^{kl}$ must be covered by $b_{ij} y_{ij}$. Now, we generalize the 3-Criteria inequality using this consideration. First, we consider inequalities that have the same right-hand side as the 3-Criteria Inequality. The right-hand side generates paths that pass through the vertices of V_2 and V_3 exactly once. Additionally, assume that all coefficients of $\bar{f}$ on the left-hand side are nonnegative. Then, the coefficients on the left-hand side must be either 0 or 1, and we must generate disjoint paths.

Check the condition in which $\mathcal{P}(k_1, l_1, i_1 j_1)$ and $\mathcal{P}(k_2, l_2, i_2 j_2)$ are disjoint. The path belonging to both must pass through k_1, k_2, l_1, l_2, and $i_1 j_1$, $i_2 j_2$. For there to be no such path, either k_2 or l_2 must be included in $[i_1 + 1, j_1 - 1]$, or k_1 or l_1 must be included in $[i_2 + 1, j_2 - 1]$.

Consider a valid inequality of the form $\sum \alpha_{ij}^{kl} \bar{f}_{ij}^{kl} \leq \sum_{a \leq i < c, c \leq j \leq b} y_{ij}$ for some $1 \leq a < c \leq b \leq n$. Since the y_{ij} on the right-hand side generate non-overlapping paths, the paths generated by the $\bar{f}_{ij}^{kl}$ on the left-hand side must also not overlap. In the 3-Criteria inequality, it was guaranteed that these f's did not overlap by satisfying $i \leq t < j$ for some t. However, the paths generated by both $\bar{f}_{ij}^{kl}$ and $\bar{f}_{op}^{qr}$ must include node k, l, q, r, and arc ij, and qr simultaneously.

Proposition 2.3.6. *$\bar{f}_{i_1 j_1}^{k_1 l_1}$ and $\bar{f}_{i_2 j_2}^{k_2 l_2}$ generate disjoint set of path if and only if k_1 or $l_1 \in [i_2+1, j_2-1]$, or k_2 or $l_2 \in [i_1+1, j_1-1]$.*

Remark 2.3.7. The class of graphs that arise when we map each of the pairs (k_1, i_1, j_1, l_1) and (k_2, i_2, j_2, l_2) to nodes and connect the pair of node if they generate disjoint set of path by an edge includes a class of graphs called interval graphs (Lekkeikerker & Boland 1962). An interval graph is an undirected graph in which each node is associated with an interval, and two nodes are connected by an edge if and only if their corresponding intervals have a nonempty intersection.

Since the set of paths generated by $\bar{f}_{ij}^{kl}$ on the left-hand side must be covered by the right-hand side, there must be elements of both V_2 and V_3 among k, i, j, and l. This can be divided into three cases: (k, i), (i, j), or (j, l) belongs to A_{23}. The 3-Criteria Inequality only considers the case where every (j, l)(or (k, i) symmetrically) of f belongs to A_{23}, but now we examine the cases where two or all three of (k, i), (i, j), and (j, l) belong to A_{23}.

Proposition 2.3.8. *Let $\phi : V_2 \to V_3$ be a nonincreasing function, S_1 be a subset of $\{(k, i, j, l) |\ 1 \le k \le i,\ i \in V_1,\ j \in V_2,\ l \in V_3\}$, and S_2 be a subset of $\{(k, i, j, l) | k \in V_2, i \in V_3, j \in V_4, j \le l \le n\}$. The inequality*

$$\sum_{(k,i,j,l) \in S_1} \bar{f}_{ij}^{kl} + \sum_{(k,i,j,l) \in S_2} \bar{f}_{ij}^{kl} \le \sum_{ij \in A_{23}} y_{ij} \tag{2.17}$$

is valid if each $ij \in A_{12}$ satisfies at most one element in S_1 such that $\phi(j) < l$, and each $ij \in A_{34}$ satisfies at most one element in S_2 such that $i \le \phi(k)$.

Proof: If $(k, i, j, l) \in S_1$, then $(j, l) \in A_{23}$ and if $(k, i, j, l) \in S_2$, then $(k, i) \in A_{23}$, so each generates a path set that is covered by the right-hand side. Now, show that

each generates disjoint paths. Elements belonging to S_1 and those belonging to S_2 generate paths that contain only distinct $ij \in A_{12}$ and $ij \in A_{34}$, respectively, so they generate disjoint paths. Now, for arbitrary $(k_1, i_1, j_1, l_1) \in S_1$ and $(k_2, i_2, j_2, l_2) \in S_2$, if $k_2 < j_1$, then $k_2 \in V_2$ and $i_1 \in V_1$, so the paths they generate are disjoint as they satisfy $k_2 \in [i_1+1, j_1-1]$. If $k_2 \geq j_1$, then by the condition, $\phi(j_1) < l_1$ and $i_2 \leq \phi(k_2)$. Since ϕ is nonincreasing, $\phi(k_2) \leq \phi(j_1)$, so we have $i_2 < l_1$. As $i_2, l_1 \in V_3$ and $j_2 \in V_4$, we have $l_1 \in [i_2 + 1, j_2 - 1]$, so the paths they generate are also disjoint. Therefore, since all elements generate disjoint paths and each path set generated is covered by the right-hand side, the given inequality is valid. □

For all $(k_1, i_1, j_1, l_1) \in S_1$ and $(k_2, i_2, j_2, l_2) \in S_2$, if there exist some $t_1 \in V_2$ and $t_2 \in V_3$ such that $j_1 \leq t_1$, then $l_1 < t_2$ and if $k_2 \leq t_1$, then $i_2 < t_2$ hold on the right-hand side even if we subtract $\sum_{ij \in A_{23}, i \leq t_1, t_2 \leq j} y_{ij}$ from it. Therefore, the corresponding inequality is still valid.

Proposition 2.3.9. *Given a non-increasing function $\phi_1 : V_1 \to V_3$ and a non-decreasing function $\phi_2 : V_2 \to V_3$, and sets $S_1 \subset \{(k, i, j, l) |\ 1 \leq k \leq i,\ i \in V_1,\ j \in V_2,\ l \in V_3\}$ and $S_2 \subset \{(k, i, j, l) |\ 1 \leq k \leq i,\ i \in V_2,\ j \in V_3,\ j \leq l \leq n\}$, the inequality*

$$\sum_{(k,i,j,l) \in S_1} \bar{f}_{ij}^{kl} + \sum_{(k,i,j,l) \in S_2} \bar{f}_{ij}^{kl} \leq \sum_{ij \in A_{23}} y_{ij} \tag{2.18}$$

is valid if for each $ij \in A_{12}$, there is at most one element in S_1 with $l \leq \min\{\phi_1(i), \phi_2(j)\}$, and for each $ij \in A_{23}$, if $k \in V_1$, then $\max\{\phi_1(k), \phi_2(i)\} < j$, and if $k \in V_2$, then $\phi_2(k) < l$ for at most one element in S_2.

Proof: Similar to the proof of Proposition 2.3.8, the elements in S_1 and S_2 gener-

ate paths that are disjoint from each other, and are covered by the given inequality's right-hand side. Consider two elements $(k_1, i_1, j_1, l_1) \in S_1$ and $(k_2, i_2, j_2, l_2) \in S_2$. Since $l_1 \leq \min\{\phi_1(i_1), \phi_2(j_1)\}$ and ϕ_1, ϕ_2 are nonincreasing and nondecreasing, respectively, we have $k_2 \leq i_1$, $j_1 \leq i_2$ implies $l_1 \leq \min\{\phi_1(i_1), \phi_2(j_1)\} \leq \max\{\phi_1(k_2), \phi_2(i_2)\} < j_2$. Likewise, $j_1 \leq k_2$, $j_1 \leq i_2$ implies $l_1 \leq \phi_2(j_1) \leq \phi_2(k_2) < j_2$. Therefore, either $i_1 < k_2 < j_1$ or $i_2 < j_1$ or $l_1 < j_2$, which means the paths generated by the two elements are disjoint. □

If only $\bar{f}$ with positive coefficients exist on the left-hand side, the right-hand side must cover the set of paths generated by each $\bar{f}_{ij}^{kl}$ with positive coefficients. To achieve this, we need to consider the conditions for the minimal set that the arc index with positive coefficient of right-hand side must include. Let first find which y_{pq} generate at least not disjoint path with the set of path that $\bar{f}_{ij}^{kl}$ generate. y_{pq} with p or q in $[i+1, j-1]$, or one of k, l, i, j is contained in $[p+1, q-1]$ is meaningless. Therefore, p and q must both belong to $[1, k], [k, i], [j, l]$, or $[l, n]$ simultaneously, or $pq = ij$. If y_{ij} is present on the right-hand side, then $p(k, l, ij) \in p(ij)$, so that itself is a minimal set. If not, assume that p and q both belong to $[1, k], [k, i], [j, l]$, or $[l, n]$. Then (p, q) can be divided into four sets that belong to $[1, k], [k, i], [j, l]$, and $[l, n]$, respectively. One of these four sets must cover the set of paths generated by $\bar{f}_{ij}^{kl}$; otherwise, y_{pq} in $[s, t]$ generates only paths that use the arc pq. If y_{pq} fails to cover a path generated by $\bar{f}_{ij}^{kl}$, it means that it did not cover the $s-t$ subpath of that path. Therefore, if the set of y_{pq} belonging to each $[1, k], [k, i], [j, l]$, and $[l, n]$ cannot cover the cases where they are in the form of $p_1, \cdots, p_4$, corresponding to $1-k$, $k-i$, $j-l$, and $l-n$ subpaths, respectively, it cannot cover the path p_1-p_2-ij-p_3-p_4. Then, to determine when the sum of y_{pq} in $[s, t]$ covers all paths that pass through all s

and t, we examine the condition for $s-t$ cuts, which are known to be $s-t$ cuts by the max-flow min-cut theorem (Dantzig & Fulkerson 2003). Therefore, we define the *minimal cover* of $\sum_{(k,l,i,j)\in S} f_{ij}^{kl}$ as the set of $\sum_{(i,j)\in S'} y_{ij}$ that cover the paths generated by $\sum_{(k,l,i,j)\in S} f_{ij}^{kl}$, where removing any y_{kl} from this set will result in a failure to cover. We call $s-t$ cut minimal $s-t$ cut, if corresponding $s \in S$ and $t \in T$ are contained in $[s,t]$.

Proposition 2.3.10. *Minimal cover of $\bar{f}_{ij}^{kl}$ is y_{ij} or $\sum_{pq\in C} y_{pq}$ where C is one of minimal $1-k$, $k-i$, $j-l$, $l-n$ cuts.*

Using observations so far, we can obtain the facet-defining condition of (2.19), which is a generalization of the every valid inequalities so far.

Theorem 2.3.11. ***Maximal Disjoint Inequality***

Given $F \subset \{(k,i,j,l)|1 \le k \le i < j \le l \le n\}$, and $1 < t_1 < t_2 \le t_3 < n$,

$$\sum_{(k,i,j,l)\in F} \bar{f}_{ij}^{kl} \le \sum_{i\in[t_1,t_2),j\in[t_2,t_3]} y_{ij} \tag{2.19}$$

is a facet-defining inequality if and only if F and (t_1,t_2,t_3) satisfy the following:

(i) For every $(k,i,j,l) \in F$, $\{k,i,j,l\} \cap [t_1,t_2) \neq \varnothing$ and $\{k,i,j,l\} \cap [t_2,t_3] \neq \varnothing$,

(ii) for every two different elements $(k_1,i_1,j_1,l_1),(k_2,i_2,j_2,l_2) \in F$, $\{k_2,i_2,j_2,l_2\} \cap [i_1+1,j_1-1]$ or $\{k_1,i_1,j_1,l_1\} \cap [i_2+1,j_2-1]$ is nonempty,

(iii) there exists $(k,i,j,l) \in F$ with (k,i), or $(j,l) = (t_1,t_3)$,

(iv) For any $(k_1,i_1,j_1,l_1) \notin F$ with ki or ij or $jl \in A_{23}$, there exists $(k_2,i_2,j_2,l_2) \in F$ with $k_1,i_1,j_1,l_1 \notin [i_2+1,j_2-1]$, $k_2,i_2,j_2,l_2 \notin [i_1+1,j_1-1]$, and $d(k_2,l_2) < U$.

Proof: (**Necessity**)

- Condition (i) is equivalent to the statement that the set of paths generated by $\bar{f}$ is covered by the right-hand side. Therefore, this is a necessary condition.

- Condition (ii) is equivalent to that each path generated by $\bar{f}_{ij}^{kl}$ is disjoint, by Proposition 2.3.6. Since the right-hand side does not generate more than one path, this is also a necessary condition.

- Condition (iii) states that if there is no element of F with (k, i) or $(j, l) = (t_1, t_3)$, the only case where the path generated by arc t_1t_3 can overlap is when there exists an element of F with $(i, j) = (t_1, t_3)$. If there is no such element in F, then subtracting $y_{t_1t_3}$ from the right-hand side is still valid, so (2.19) is not facet-defining. Let us consider the case where there is an element (k', t_1, t_3, l') in F with $(i, j) = (t_1, t_3)$. Removing this element means there is no term on the left-hand side that generates the path t_1t_3. Therefore, in this case, even if we subtract $\bar{f}_{t_1t_3}^{k'l'}$ from the left-hand side and $y_{t_1t_3}$ from the right-hand side, the inequality (2.19) is still valid. Since $\bar{f}_{t_1t_3}^{k'l'} \leq y_{t_1t_3}$, we can conclude that (2.19) is not facet-defining.

- Condition (iv) means that for any $(k_1, i_1, j_1, l_1) \notin F$ with ki or ij or $jl \in A_{23}$, there exists $(k_2, i_2, j_2, l_2) \in F$ such that $\bar{f}_{i_2j_2}^{k_2l_2} = d(k_2, l_2)$ and at the same time $\bar{f}^{k_1l_1}i_1j_1 = 1$ can be true. Let us call such an element of F a corresponding element. If there are multiple corresponding elements, we choose one of them arbitrarily as the corresponding element. If there is no corresponding element, then $\bar{f}_{i_1j_1}^{k_1l_1}$ is always zero in the case where the equality in (2.19) holds. Since the right-hand side must be zero when its value is zero, and when the right-

Table 2.3: Affinely independent vectors with $0 = 0$

new variable	quantifier	path	flow
y_{1n}		$1-n$	
y_{ij}	$1 \le i < j < n,\ ij \notin A_{23}$	$1-i-j-n$	
f_{kl}^{kl}	$1 \le k < l \le n,\ kl \notin A_{23}$	$1-k-l-n$	$x_{kl} = 1$
f_{kj}^{kl}	$1 \le k < j \le l \le n,\ kj, jl \notin A_{23}$	$1-k-j-l-n$	$x_{kl} = 1$
f_{ij}^{kl}	$1 \le k \le i < j \le l \le n,\ ki, ij, jl \notin A_{23}$	$1-k-i-j-l-n$	$x_{kl} = 1$

hand side is one, there must be a term on the left-hand side such that $\bar{f}_{ij}^{kl} = 1$, (2.19) remains valid even if we add $\bar{f}_{i_1 j_1}^{k_1 l_1}$ to the left-hand side. This means that (2.19) is not facet-defining.

(**sufficiency**) If both the first and second conditions are satisfied, we can see (2.19) becomes a valid inequality.

Now, we will find affinely independent vectors of the same number as $dim(P)$. As in Theorem 2.3.2, we will list the vectors, and each listed vector will have non-zero components for the first time. Let the name of each feasible solution vector be $v(x)$, where x is the variable that first becomes non-zero in that vector. First, let's list the vectors for the variables that can be non-zero Even if a path does not pass through a vertex belonging to $V_2 := [t_1, t_2 - 1]$ or $V_3 := [t_2, t_3]$. Let $A_{23} = ij | i \in V_2, j \in V_3$.

Table 2.3 exhibits the maximal affinely independent vectors when both sides are 0. Now, we will list the vectors when both sides are 1.

For each $(k, i, j, l) \in F$, $v(\bar{f}_{ij}^{kl})$ satisfies the conditions: the path is $1-k-i-j-l-n$ and the flow is $x_{kl} = d(k, l)$.

For the third condition, $(k_2, i_2) = (t_1, t_3)$ in (k_2, i_2, j_2, l_2) and $(j_1, l_1) = (t_1, t_3)$ in (k_1, i_1, j_1, l_1) cannot satisfy condition ii). Without loss of generality, assume that $(k^*, i^*, t_1, t_3) \in F$ exists.

When $i \in V_2$, $j \in V_3$, and $(i,j) \neq (t_1,t_3)$, $v(y_{ij})$ satisfies the conditions: the path is $1 - k^* - i^* - t_1 - i - j - t_3 - n$ and the flow is $x_{k^*t_3} = d(k^*,t_3)$.

If $(k,i,j,l) \notin F$ and ki or ij or $jl \in A_{23}$, then $v(\bar{f}_{ij}^{kl})$ satisfies the conditions: the path includes 1, n, k, k', i, i', j, j', l, and l', and the flow is $x_{k'l'} = d(k',l')$ and $x_{k'l'} = 1$. Here, $(k',i',j',l') \in F$ corresponds to the element in condition (iv).

We have provided vectors corresponding to the variables excluding y_{in} $(1 < i < n)$, $y_{t_1t_3}$, and f_{il}^{kl} $(1 \leq k \leq i < l \leq n)$. $y_{t_1t_3}$ corresponds to (2.2) with $i = 1$, y_{in} $(1 < i < n)$ corresponds to (2.2) with $1 < i < n$, and f_{il}^{kl} corresponds one-to-one to (2.3), so we found $dim(P(AF))$ affinely independent vectors. Thus, (2.19) becomes a facet-defining inequality. □

Valid Inequalities for Capacitated Case

We now explore capacitated sMTP. First, look at the 3-Criteria Inequality. The left-hand side of (2.13) has terms in the form of $f_{ij}^{kl}/d(k,l)$ for each ij, where $l \in V_3$ and the value is divided by $d(k,l)$ so that it is maximized at 1. If capacity U exists, a valid option is to replace it with $\sum_{1 \leq k \leq i,j \in V_3} f_{ij}^{kl}/U$. However, this is subject to the constraint (2.4), which limits it to be no greater than yij. Note that if $\sum_{k \leq i,j \leq l} d(k,l) < U$, the inequality using this term becomes redundant due to inequalities using the term $f_{ij}^{\pi(ij)}/d(\pi(ij))$.

Proposition 2.3.12. *For $1 \leq t_1 < t_2 < t_3 < n$, let $V_1 = [1,t_1]$, $V_2 = [t_1+1,t_2]$, $V_3 = [t_2+1,t_3]$, and $V_4 = [t_3+1,n]$. Define $A_{ij} = V_i \times V_j$, $\forall\ 1 \leq i < j \leq 4$.*

For $(i,j) \in A_{12}$, let $S(i.j) = \{f_{ij}^{kl}/d(k,l) | 1 \leq k \leq i, l \in V_3\} \cup \{\sum_{k \leq i,j \leq l} f_{ij}^{kl}/U | \sum_{k \leq i,j \leq l} d(k,l) > U\}$.

The following inequality is valid for any selection of each element s_{ij} from $S(i,j)$:

$$\sum_{(i,j)\in A_{12}} s_{ij} \leq \sum_{(i,j)\in A_{23}} y_{ij}. \tag{2.20}$$

Note this idea can also be applied to Propositions 2.3.8 and 2.3.9 to generalize the conditions for their valid inequality.

2.4 Triple Formulation

2.4.1 Triple Formulation and Basic Properties

Triple formulation (TF) is adapted from a compact representation called Triple formulation that has been sucessfully applied to the Maximum Concurrent Flow Problem and Backhaul Profit Maximimzation Problem (Dong et al. 2014, 2022). The Arc-Flow Formulation is easy to understand in terms of representation, but it consists of $O(n^4)$ variables and $O(n^4)$ constraints, making it large in size. Without tightening, the number of constraints decreases to $O(n^3)$. On the other hand, the Triple Formulation introduced in this subsection consists of $O(n^3)$ variables and $O(n^2)$ constraints. (TF), as we will see later, has the advantage of having a shorter time to solve the LP-relaxation due to its size, even though the LP-relaxation is looser than that of (AF).

u^l_{ij} represents the amount of commodities sold at l passing through arc ij, and for simplicity, $\theta_{ij} = \sum_{l:j\leq l} u^l_{ij}$, the total amount of commodities passing through arc ij, is also introduced. The Triple Formulation is expressed as follows.

Problem 2.4.1.

$$\max \quad \sum_{1\le i<j\le n} (r(i,j)x_{ij} - c_{ij}\theta_{ij}) \tag{2.21}$$

$$\text{s.t.} \quad \sum_{j:i<j\le n} y_{ij} - \sum_{j:1\le j<i} y_{ji} = \begin{cases} 1, & i=1 \\ 0, & 1<i<n \end{cases}, \quad i \in V, \tag{2.22}$$

$$\sum_{j:k<j\le l} u^l_{kj} - \sum_{i:i<k} u^l_{ik} = x_{kl}, \qquad 1 \le k < l \le n, \tag{2.23}$$

$$\sum_{j\le l} u^l_{ij} = \theta_{ij}, \qquad 1 \le i < j \le n, \tag{2.24}$$

$$\theta_{ij} \le U y_{ij}, \qquad 1 \le i < j \le n, \tag{2.25}$$

$$x_{kl} \le d(k,l), \qquad 1 \le k < l \le n, \tag{2.26}$$

$$y_{ij} \in \{0,1\},\ 1 \le i < j \le n,\ x_{ij} \ge 0,\ u^l_{ij} \ge 0, \quad 1 \le i < j \le l \le n. \tag{2.27}$$

Let P(TF) denote the polyhedron defined by the constraints in the formulation, excluding the integer condition, and refer to this space as the (u, y)-space. Constraint (2.22) is equivalent to constraint (2.2). Constraint (2.23) corresponds to constraint (2.3). Constraint (2.25) is a capacity constraint, and constraint (2.26) is a constraint that sets an upper bound on the demand. The fundamental reason why the problem can be modeled using the Triple-variable u instead of the flow-variable f is that the objective function can be expressed in terms of u. This is because the cost per unit flow imposed on each commodity is the same for each lane.

Proposition 2.4.2. *(TF) is a valid formulation for sMTP.*

Proof: Consider an integer solution. Let the corresponding path be denoted as $p = v_1 - v_2 - \cdots - v_k$. Then, according to constraint (2.25), only u^l_{ij} where $ij, l \in p$ can be positive. Therefore, (2.23) implies that $u^l_{v_1 v_2} = x_{v_1 l}$, and for each $1 \le i \le k-2$,

$-u^l_{v_iv_{i+1}} + u^l_{v_{i+1}v_{i+2}} = x_{v_{i+1}l}$. Thus, we have $u^l_{v_iv_{i+1}} = \sum_{j=1}^{i} x_{il}$ as intended. □

Proposition 2.4.3. *P(TF) ⊃ P(AF).*

Proof: If each u^l_{ij} in the constraint of (TF) is replaced with $\sum_{k\leq i} f^{kl}_{ij}$, then the constraint (2.22) is equivalent to constraint (2.2), and constraint (2.25) is equivalent to constraint (2.4). Finally, for each (k, l) satisfying $1 \leq k < l \leq n$, constraint (2.23) becomes the same as the constraint of (2.3) when $k = i < l$, and the constraint of (2.3) with $k < i < l$ becomes an identity if $u^l_{ij} = \sum_{k\leq i} f^{kl}_{ij}$. □

Proposition 2.4.4. *dim(TF)* $= \frac{(n-1)(n-2)}{2} + \frac{(n-1)n(n+1)}{6}$

Proof: The u-variable has $(n-1)n(n+1)/6$ elements, and the y-variable has $n(n-1)/2$ elements. Just like when checking the dimension of (AF), we can obtain $(n-1)(n-2)/2+1$ affinely independent vectors with $u = 0$. By using $i-j$ flow on the $1-i-j-n$ path, we can find vectors where only u^j_{ij} is positive. By using $i-l$ flow on the $1-i-j-l-n$ path, we can find vectors where only one of the u^l_{ij} satisfying $j < l$ is positive. We have $(n-1)(n-2)/2 + (n-1)n(n+1)/6 + 1$ vectors, which is the same as the one plus the total number of variables minus $n-1$ equations. □

The conditions for the valid inequality of (TF) can also be obtained, similar to (AF). The constraints of the extended formulation, excluding the constraint of (2.22) and the non-negativity constraint, are as follows.

$$
\begin{aligned}
-\sum_{p\in P, k:ij\in p, k\le i, k,l\in p} z_p^{kl} &\le -u_{ij}^l, && 1\le i<j\le l\le n, && (2.28a)\\
\sum_{p\in P, k:ij\in p, k\le i, k,l\in p} z_p^{kl} &\le u_{ij}^l, && 1\le i<j\le l\le n, && (2.28b)\\
z_p^{kl} - d(k,l)z_p &\le 0, && 1\le k<l\le n, r(k,l)\in P && (2.28c)\\
\sum_{k,l:1\le k\le i<l\le n, k,l\in p} z_p^{kl} - Uz_p &\le 0, && 1\le i<n, p\in\mathcal{P} && (2.28d)\\
\sum_{p\in P:ij\in p} z_p &\le y_{ij}, && 1\le i<j\le n. && (2.28e)
\end{aligned}
$$

From this, we can obtain the proposition below.

Proposition 2.4.5. *In (u,y)-space, the only form of facet-defining inequality, aside from the constraints of (TF), is as follows:*

$$\sum a_{ij}y_{ij} \ge \sum b_{ij}^l u_{ij}^l - \sum c_{ij}^l u_{ij}^l \ \ (a,b,c\ge 0)$$

The necessary condition for this inequality to be a facet-defining inequality and the sufficient condition for it to be a valid inequality are that the following must hold for each 1-n path p:

$$\sum_{ij\in p} a_{ij} \ge \max_{\mu_i\ge 0}(U\sum_{i=1}^{n-1}\mu_i + \sum_{k,l\in p} d(k,l)\max\{\sum_{ij\in p}(b_{ij}^l - c_{ij}^l) - \sum_{i=k}^{l-1}\mu_i, 0\})$$

2.4.2 3-Criteria-TF Inequality

In this subsection, we introduce a family of valid inequalities for TF called 3-Criteria-TF.

First, considering (2.26) in problem 2.4.1, we can establish the following set of inequalities for each $k-l$ cut $(S,\bar{S})$ $(k\in S\subset[k,l-1],; l\in\bar{S}\in[k+1,l])$, based

on the fact that x_{kl} can only have a positive value when it passes through vertices k and l.

$$x_{kl} \leq d(k,l) \left(\sum_{i \in S, j \in \bar{S}} y_{ij} \right)$$

By computing the $k-l$ min-cut for each (k,l) pair, we can perform separation for the entire set of inequalities. However, this separation process is time-consuming, and in most cases, the inequalites are not facet-defining. Therefore, we aim to consider only a subset of these cuts. Firstly, we can replace the constraint (2.26) of problem 2.4.1 with two inequalities when $S = k$ or $\bar{S} = l$. By doing so, we can obtain a tighter formulation while maintaining the number of constraints in the problem at $O(n^2)$.

$$x_{kl} \leq d(k,l)(\sum_{k<j\leq l} y_{kj}) \tag{2.29a}$$

$$x_{kl} \leq d(k,l)(\sum_{k\leq i<l} y_{il}) \tag{2.29b}$$

We can also consider incorporating additional $O(n^3)$ constraints below.

$$x_{kl} \leq d(k,l)(\sum_{k\leq i\leq t<j\leq l} y_{ij}) \tag{2.30}$$

However, the inequality (2.30) does not define a facet when $k < t$, rather than inequality (2.29a). When equality holds, for (i,j,l') where $i < k < j \leq t$ and $t < l' \leq l$, we can verify that $u_{ij}^{l'}$ is always 0. If $u_{ij}^{l'} > 0$, the right-hand side of (2.30) should be $d(k,l)$, but the left-hand side of (2.30) cannot be positive since it is not possible that a path pass through the arc ij and the node k simultaneously. Taking this into

account, we can derive the following inequality similar to 3-Criteria inequality (2.13) for any $1 \leq k \leq t < l \leq n$ and any function $\pi : [1,k] \times [k+1,t] \to [t+1,l]$. Let's refer to the inequalities as 3-Criteria-TF.

$$\sum_{i \leq k, k+1 \leq j \leq t} \frac{d(k,l)}{\sum\limits_{k' \leq k} d(k', \pi(i,j))} u_{ij}^{\pi(i,j)} + x_{kl} \leq d(k,l)(\sum_{k \leq i \leq t < j \leq l} y_{ij}) \tag{2.31}$$

Proposition 2.4.6. *3-Criteria-TF inequalities are valid for TF.*

Proof: Similar to the 3-Criteria inequality, considering the left-hand side of the equation, where $u_{ij}^{\pi(i,j)}$ represents the edge ij traversing the vertices $1 \leq i \leq k$ and $k+1 \leq j \leq t$, and x_{kl} represents the node k being traversed, we observe that only one term can have a positive value at a time. In the case where only one term exists, we can confirm its validity by verifying that $x_{kl} \leq d(k,l)$ and $u_{ij} \leq \sum\limits_{k' \leq k} d(k', \pi(i,j))$.
□

Algorithm 2.4.7. ***Separation algorithm for 3-Criteria Inequality***

1. *For each $1 \leq i < c \leq b < n$, compute $y_{i[c,b]} := \sum\limits_{j=[c,b]} y_{ij}$. Since $y_{i[c,b-1]} + y_{ib}$ has the same value, each can be calculated in $O(1)$ time, so the overall computation time is $O(n^3)$.*

2. *For each $1 < a < c \leq b < n$, compute $y_{[a,c,b]} := \sum\limits_{i \in [a,c-1]} y_{i[c,b]}$. This can be calculated with an operation $y_{[a-1,c,b]} + y_{a[c,b]}$, so the overall computation time is $O(n^3)$.*

3. *For each $1 \leq i < j \leq t < l < n$, compute $u_{ij}^{[t,l]} := \max\limits_{l' \in [t+1,l]} u_{ij}^{l}$. This*

can be calculated with an operation by $\max\{u^l_{ij}, u^{[t+1,l-1]}_{ij}\}$*, so the overall computation time is* $O(n^4)$*.*

4. *For each* $1 \leq i < j \leq c \leq b < n$*, compute* $u^{[k,t,l]}_j := \sum i < k u^{[t,l]}_{ij}$*. This can be calculated with an operation by* $u^{[k-1,t,l]}_j + u^{[t,l]}_{(k-1)j}$*, so the overall computation time is* $O(n^4)$*.*

5. *For each* a*, compute* $u^{[k,t,l]} := \sum_{j \in [k+1,t]} u^{[k,t,l]}_j$*. Each takes* $O(n)$ *time, so the overall computation time is* $O(n^4)$*.*

6. *Finally, for each* $1 \leq a \leq c < b < n$*, compute the maximum value of* $u^{[k,t,l]} + x_{kl} - y_{[k,t,l]}$*, which represents the maximum violation of the 3-Criteria-TF Inequality. This can be computed in* $O(n^3)$ *time.*

2.4.3 Other Valid Inequalities for the Triple Formulation

Now we examine another family of valid inequalities. Let $[i, j]$ be a set of nodes from i to j, and let $d(S, T) = \sum_{s \in S, t \in T} d(i, j)$. Then, we have

$$u^l_{ij} \leq d([1, i], l) y_{ij}, \ \forall\ 2 \leq i + 1 < j \leq l \leq n. \tag{2.32}$$

Proposition 2.4.8. *(2.32) are facet defining inequality if and only if* $d([1, i], l) < U$*.*

Proof: Let ε be a positive number less than or equal to $U - d([1, i], l)$. Let R be the set of solutions that have paths in the form of 1-n, 1-a-n, 1-a-b-n $(1 < a < b < n)$ and do not have a path of the form i-j for $(a, b) \neq (i, j)$ and $(b, c) \neq (i, j)$ and $1 \leq a < b \leq c \leq n$. Let S be the set of solutions that have a path passing through nodes a, b, l and a flow of $a - l$ with a value of ε, for (a, b, c) as defined above. Let T be the set of solutions that have a path of the form $1 - 2 - \cdots - (i - 1) - i - j - l - n$

and $a-l$ flow with a value of $d(a,l)$ for $1 \leq a \leq i$. Let U be the set of solutions that have a path passing through node $j \leq l' \neq l$ and $i-l'$ flow with a value of ε. Finally, let V be the set of solutions that have $a-j$ flow with a value of ε for $1 \leq a < i$ and a path passing through nodes $1, a, i, i+1, j, n$. Then the sets R, S, T, U, and V are affinely independent and the total number of solutions is $1+(n-1)(n-2)/2+(n+1)n(n-1)/6-1 = dim(TF)$. $\square$

This is relevant only if there exists an $i-l$ flow, there exists an $a-l$ flow for all a with $a < i$. This can be complemented by the following inequality.

$$u_{ij}^l - \sum_{k<a<i} u_{ai}^l \leq d([1,k] \cup i, l) y_{ij}, \ \forall 1 \leq a < i < j \leq l \leq n. \tag{2.33}$$

To generalize this, we can consider the following inequality, which includes the previous (2.32) and (2.33), and by showing the validity of this inequality, we can prove the validity of (2.32) and (2.33).

For $S \subseteq [1, i-1]$ and $T = [1, i] \setminus S$, we can consider the following inequality:

$$u_{ij}^l - \sum_{s \in S, t \in T} u_{st}^l \leq d(T, l) y_{ij} \tag{2.34}$$

Proposition 2.4.9. *(2.32-2.34) are valid.*

Proof: If the arc ij is not used, the right-hand side is 0, and the left-hand side is obtained by subtracting a nonnegative value from 0. If the arc ij is used, all flows starting from any i before l and arriving at l must pass through node i. That is, there is no direct flow from S to a node with an index larger than i, so the left-hand side is greater than or equal to the total sum of flows starting from an element

in S and arriving at l, which can be expressed as $\sum_{s \in S, t \in T} u^l_{st}$. The left-hand side is obtained by subtracting u^l_{ij} from this value, and it is less than the total sum of flows starting from a node not in S before i and arriving at l. This total sum is equal to $d([1, i] \setminus S, l)$, so (2.34) holds. $\square$

(2.34) can be generalized by the following inequalities for $J \subset [i+1, l]$.:

$$\sum_{j \in J} u^l_{ij} - \sum_{s \in S, t \in T} u^l_{st} \leq d(T, l) \sum_{j \in J} y_{ij} \tag{2.35}$$

We call (2.35) as Cutset Inequality. The validity of Cutset Inequality can be confirmed from the proof that (2.34) is valid. When using arc ij as an argument for each $j \in J$, it can be observed that the right-hand side is equal to zero and the left-hand side is non-positive even when arc ij is not used for all $j \in J$, using the same argument as when arc ij is used for each $j \in J$.

(2.32) and (2.33) have $O(n^3)$ and $O(n^4)$ constraints, respectively. Therefore, it is possible to add them to the (TF) from the beginning. However, since (2.34) and (2.35) is exponentially many, it needs to be added adaptively to the set of constraints using a separation algorithm. We propose ⊡ separation algorithm in (2.35). Although we can obtain the separation algorithm in (2.35) from this, there is no particular benefit in terms of computation time.

Algorithm 2.4.10. ***Separation algorithm for (2.35)***

Repeat the following for each $1 \leq i < l$*:*

1. *Create a graph* $G(i, l)$ *by adding a new vertex* 0 *and* $\{i+1, \cdots, l\}$ *to the induced subgraph of original graph defined by vertices* $1, \cdots, i$*. The arc is*

composed of $A_1 = \{ab | 0 \leq a < b \leq i\}$ *and* $A_2 = \{ba | i+1 \leq b \leq l, 1 \leq a \leq i\}$.

2. *Let the capacity of arcs* ab *in* A_1 *be* u^l_{ab}, *and* ba *in* A_3 *be* $d(a,l)y_{ib}$ *if* $a \neq i$ *and* $d(i,l)y_{ib} - u^l_{ib}$ *if* $a = i$.

3. *Compute the* $0-i$ *min-cut and check if it is smaller than* 0. *If it is smaller than* 0, *the current solution is cut-off by the inequality defined in (2.35), which has* S *as the set of nodes of* $[1,i]$ *on the same side with* 0 *and* J *as the set of nodes of* $[i+1,l]$ *on the same side with* 0, *in the min-cut.*

Proposition 2.4.11. *Algorithm 2.4.10 is a polynomial-time separation algorithm for (2.35).*

Proof: (2.35) can be rewritten as follows:

$$0 \leq \sum_{s \in S, t \in T} u^l_{st} + d(T,l) \sum_{j \in J} y_{ij} - \sum_{j \in J} u^l_{ij}$$

The value on the right-hand side is exactly the capacity of the cut $S(\{0\} \cup S, T)$ in the graph constructed in Steps 1 and 2 in the Algorithm 2.4.10. Since the algorithm requires only solving min-cut $O(n^2)$ times after the construction of the graph, it takes polynomial time. $\square$

Chapter 3

Approximation Algorithm

In this chapter, we present research on the approximability of sMTP. We demonstrate that the integrality gap of the formulations presented in Chapter 2 is $\theta(n^2)$, and that sMTP is $2^{\log^{1-\varepsilon} n}$-hard to approximate for any $\varepsilon > 0$, unless NP can be solved in quasi-polynomial time. We present approximation algorithms for ssMTP, t-separable sMTP, and an approximation algorithm for sMTP that uses approximation algorithm for sMTP and t-separable sMTP as subroutines. We also provide approximation algorithms for other problems related to sMTP, including TRPP.

3.1 Introduction

Approximation algorithms are efficient algorithms that aim to guarentee a certain level of performance even in the worst case.

Definition 3.1.1. We call an polynomial time algorithm α-*approximation* that outputs a feasible solution if the following always holds. We call α the *approximation ratio* of the algorithm.

$$\frac{\text{optimal objective value}}{\text{objective value of the solution returned by the algorithm}} \leq \alpha$$

The development of such approximation algorithms is important for the following

reasons (Williamson & Shmoys 2011):

- It offers a solid mathematical foundation for the examination of heuristics.
- When designing algorithms, the initial emphasis is often placed on idealized models rather than their practical application in the real world.
- It provides a measure to quantify the difficulty of the optimization problems.

Furthermore, in cases where there is no information available regarding the realism of a given instance, or in other words, when there is no information about the probability distribution of the instance, this can be considered the only method that guarantees the quality of the solution to the problem. This signifies an approximation algorithm with good ratio for a problem is a favorable choice to consider as a potential solution when encountering the problem for the first time in a new application. Research on the inapproximability of a certain problem is a study that aims to demonstrate the non-existence of an approximation algorithm that achieves a specific ratio for that problem.

Definition 3.1.2. We say a problem is *α-hard to approximate*, or hard to α-approximate, if there cannot exists and α-approximation algorithm.

Research on inapproximability or hardness of approximation is important because it reveals the limitations that we must face when developing approximation algorithms (Arora & Lund 1996). Since the solutions output by approximation algorithms are feasible, they can be a good alternative when we are interested in worst case performance and need to compromise on time. Additionally, if the approximation ratio of an algorithm is α and and the objective function value of the solution

is $\mathrm{OPT}_{\mathrm{app}}$, we can determine that the optimal objective function value OPT of the problem belongs to the range $[\mathrm{OPT}_{\mathrm{app}}, \alpha\mathrm{OPT}_{\mathrm{app}}]$. This means that not only can we obtain a lower bound for OPT from the solution output by the approximation algorithm, but also an upper bound, and this information can be helpful in developing other types of solution methods. Of course, from the MILPs obtained in Chapter 2, we can also obtain an upper bound for the objective value of the LP relaxation obtained by relaxing the integer constraints of the variables, which is denoted as $\mathrm{OPT}_{\mathrm{LP}}$. If we know the value of the integrality gap defined below, we can use it to obtain a lower bound.

Definition 3.1.3. The *integrality gap* of a given MILP formulation is minimum β such that the following holds.

$$\frac{\text{optimal objective value of the LP-relaxation problem}}{\text{optimal objective value}} \leq \beta$$

For example, if the integrality gap is β, we can guarantee that OPT belongs to the range $[\mathrm{OPT}_{\mathrm{LP}}/\beta, \mathrm{OPT}_{\mathrm{LP}}]$. Furthermore, since the integrality gap is no greater than the ratio of the approximation solution obtained by using LP relaxation, it becomes an important indicator for peeking at the limit of the approximation solution method using the corresponding MILP formulation, its lower bound on the ratio. One commonly used method for obtaining feasible solutions from LP relaxation is randomized rounding (Raghavan & Tompson 1987), which involves rounding the solution obtained from LP relaxation. The approximation ratio of such algorithms is the product of the ratio of the decrease in the objective function value during the rounding process to the integrality gap. For this reason, research has been conducted to find integrality gaps (Chekuri et al. 2006, Chalermsook et al. 2012).

To establish an inapproximability of sMTP, we consider the *Max-Rep* (Kortsarz 2001) which is an equivalent variant of the maximization version of the *label cover problem*, a fundamental problem in the study on the hardness of approximation of combinatorial optimization (Arora & Lund 1996).

There is no polynomial approximation algorithm for the Max-Rep with ratio $2^{\log^{1-\varepsilon} n}$, for any $0 < \varepsilon < 1$, unless NP is quasi-polynomially solvable (Arora et al. 1997). There is still a large gap between the lower bound and upper bound on the approximability of the Max-Rep. Charikar et al. (2011) presented $O(n^{1/3})$-approximation algorithm for the Max-Rep. Note that they also show that the integrality gap of a natural MILP for Max-Rep is $\Omega(\sqrt{n})$. For a special class of the Max-Rep, known as "satisfiable," Manurangsi and Manurangsi & Moshkovitz (2017), and Chlamtáč et al. (2017), respectively, attained $O(n^{1/4})$ and $O(n^{0.23})$-approximation. Peleg (2007) generalized the Max-Rep with weighted edges and nonuniform sets and proposed an $O(n^{1/2})$-approximation algorithm.

Section 3.2 covers research on the inapproximability of sMTP and the integrality gap of MILP formulations discussed in Chapter 2 from these perspectives. This can be seen as research on the inapproximability of approximation solutions using LP relaxation. In Section 3.3, we discuss approximation algorithms for the special case of sMTP, which are used as subroutines in the sMTP approximation algorithm. In Section 3.4, we present an approximation solution for sMTP by utilizing the results mentioned above. Finally, in Section 3.5, we demonstrate the extension of sMTP's approximation solutions to more general problems or cases with minor variations, including the Traveling Repairman Problem with Profit (TRPP), by utilizing the techniques used in sMTP's approximation algorithm.

3.2 Inapproximability Results

3.2.1 Integrality Gap

Let us analyze the integrality gap of the AF formulation for sMTP. the following is the AF.

$$
\begin{aligned}
\max \quad & \sum_{1\le k<l\le n}\left(r(k,l)x_{kl} - \sum_{k\le i<j\le l} c_{ij} f_{ij}^{kl}\right) \\
\text{s.t.} \quad & \sum_{j:i<j\le n} y_{ij} - \sum_{j:1\le j<i} y_{ji} = \begin{cases} 1, & i=1 \\ 0, & 1<i<n \end{cases}, \\
& \sum_{j:i<j\le l} f_{ij}^{kl} - \sum_{j:k\le j<i} f_{ji}^{kl} = \begin{cases} x_{kl}, & i=k \\ 0, & k<i<l \end{cases}, \quad 1\le k\le i<l\le n, \\
& \sum_{k\le i<j\le l} f_{ij}^{kl} \le U y_{ij}, \qquad 1\le i<j\le n, \\
& x_{kl} \le d(k,l), \qquad 1\le k<l\le n, \\
& y_{ij}\in\{0,1\},\ x_{kl}\ge 0,\ f_{ij}^{kl}\ge 0, \qquad 1\le k\le i<j\le l\le n.
\end{aligned}
$$

Theorem 3.2.1. *The integrality gap of AF is $\theta(n^2)$.*

Proof: Let an arbitrary instance is given, and consider the LP relaxation. The objective function can be partitioned into $O(n^2)$ values related to each pair $k-l$, given by $r(k,l)x_{kl} - \sum_{k\le i<j\le l} c_{ij} f_{ij}^{kl}$. The solution space remains the same, and among the optimal solutions to the problem defined by the partitioned objective function, there exists an integer solution that crosses each $k-l$ shortest path. Furthermore, the sum of the optimal objective values of each problem defined by the partitioned objective function is at least $\mathrm{OPT}_{\mathrm{LP}}$. Since each integer solution has an objective function

value less than OPT, we have $n(n-1)/2\text{OPT} \geq \text{OPT}_{\text{LP}}$. Thus, the integrality gap is $O(n^2)$. Note that the best among the feasible solutions that trade only one $k-l$ pair is an $O(n^2)$-approximation algorithm.

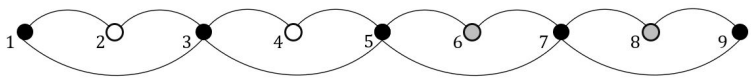

Figure 3.1: An $n = 9$ instance with large gap

considering the following instance. In this instance, n is an integer of the form $4k+1$ for some integer k. Every length of the arcs is 1, and the arcs are composed of only the arcs between $2i-1$, $2i$, and $2i+1$, for each $i \in 1, \cdots, 2k$. The s-d pairs are the all pair of nodes with even indices and exactly one of their indices is less than $2k+1$, with demand 1. The revenue of each s-d pair is equal to one plus the shortest distance between node pairs. In other words, for each $1 \leq i < k < j \leq 2k$, $r(2i, 2j) = (j-i)+2$ and $d(2i, 2j) = 1$. U is a sufficiently large value. Then, each $(2i, 2j)$ is profitable only if the path does not visit other nodes with even indices between $2i$ and $2j$. Therefore, only one s-d pair can be profitable at a time, and the optimal objective value is $((j-i)+2)-((j-i)+1) = 1$. On the other hand, let us consider a fractional solution $y_{ij} = 1/2$ for all defined arcs ij and $x_{kl} = 1/2$ for all defined s-d pairs (k, l), with f_{ij}^{kl} following the $k-l$ shortest path. It is easy to see that this solution satisfies all constraints, and the corresponding objective value is $k^2/2$. Therefore, there is a difference between the optimal objective value and $\Omega(n^2)$. $\square$

Let the optimal objective value of a sMTP instance be a and let the optimal objective value of its LP relaxation be denoted as αa. Now, consider the following

instance. Let the graph of the original instance be G with start and end nodes 1 and n, respectively. Add new nodes 0 and $n+1$ and connect them to nodes 1 and n with arcs of sufficiently large cost (greater than αa is enough). Then create an arc with cost 0 from node 0 to node $n+1$, and set $d(0, n+1) = 1$ and $r(0, n+1) = (\alpha - \varepsilon)a$ for some $\varepsilon > 0$.

If we do not use the arc $0(n+1)$, the integer solution has an objective value of at most a, while if we use it, the optimal objective value is $(\alpha - \varepsilon)a$. On the other hand, the optimal objective value of the relaxed problem is αa, which does not use the arc $0(n+1)$. Therefore, solutions that consider only the arcs with positive values in the LP relaxation and do not use the arc $0(n+1)$ have an approximation ratio of at least $\alpha - \varepsilon$. Since the integrality gap of the Triple Formulation is also loose compared to (AF), we aimed to develop an approximation method that does not rely on MILP formulation.

Corollary 3.2.2. *The integrality gap of TF is $\theta(n^2)$.*

Corollary 3.2.3. *Algorithms that output solution with $y_{ij} = 1$ only if $y_{ij} > 0$ in the LP-relaxation solutions of (AF) or (TF) for any arc ij has an approximation ratio of $O(n^2)$.*

3.2.2 Inapproximability of sMTP

In this subsection, we show that sMTP is $2^{\log^{1-\varepsilon} n}$-hard to approximate for any $\varepsilon > 0$ unless $NP \subseteq DTIME(n^{\text{poly} \log n})$. This result is obtained from that the Nonuniform Weighted Max-Rep is a special case of sMTP and inapproximability result for Max-Rep. The Nonuniform Weighted Max-Rep problem was originally defined in Peleg (2007).

Problem 3.2.4. Nonuniform Weighted Max-Rep Given a bipartite graph $G = (A, B, E)$, $A = \bigcup_{i=1}^{i=n} A_i$, $B = \bigcup_{j=1}^{j=m} B_j$, and associated nonnegative integer weight w_{ij} for each pair of sets (A_i, B_j), where $A_i = \{a_i^1, \cdots, a_i^{n_i}\}$ $\forall i \in \{1, \ldots, n\}$, $B_j = \{b_j^1, \cdots, b_j^{m_j}\}$ $\forall j \in \{1, \ldots, m\}$, choose a node from each A_i and B_j so that the sum of weights for each pair of sets whose chosen nodes are the endpoints of an edge is maximized.

We will refer to the special case of this problem as Max-Rep, where $n = m$, $n_i = m_j$, and $w_{ij} = 1$ for all $i, j \in \{1, \cdots, n\}$.

Lemma 3.2.5. *Nonuniform Weighted Max-Rep is a uncapacitated sMTP with zero arc costs and N-Max-Rep is an uncapacitated sMTP with zero arc costs and binary revenues per unit of trading.*

Proof: Given a Nonuniform Weighted Max-Rep instance $G = (A, B, E)$ and weight W, construct a sMTP instance as follows.

Let $N_k := \sum_{i=1}^{k} n_i$ and $M_l := \sum_{j=1}^{l} m_j$. Define node set $V = \{1, 2, \cdots, N_n + M_m + 3\}$ such that $N_{i-1} + j + 1$ corresponds to a_i^j, and $N_n + M_{i-1} + j + 2$ corresponds to b_i^j. Thus, nodes 1, $N_n + 2$, and $N_n + M_m + 3$ do not correspond to any node in G. Let $t = N_n + 2$. The nodes of V between 1 and t (t and $N_n + M_m + 3$) in increasing order correspond to the sets of Max-Rep in the order of $A_1, \cdots, A_n$ ($B_1, \cdots, B_m$, respectively). We connect each node $u \in V$ to every node v with $v > u$. For each $(a, b) \in E$, assign revenue 1 and demand w_{ij} to each pair of nodes of sMTP that correspond to $a \in A_i$ and $b \in B_j$. There is no other node pair in the constructed sMTP having a positive demand or revenue. We assign cost 1 to every edge if it is *internal* i.e. if its end nodes correspond to nodes from the same A_i or B_i of the Nonuniform Weighted Max-Rep for some i.

We will show that there exists an optimal solution of the constructed sMTP where the path visits exactly one node from each of the node sets corresponding to A_i and B_i for every i, and where the optimal values of the two problems are equal. Suppose a path contains an internal arc uv with cost 1. Given the order of nodes in V corresponding to A_i's and B_i's, the demand assignment rule, and the fact that the maximum revenue from an s-d pair is 1, every positive profit is achievable only on the subpath from node 1 to node u, or the subpath from node v to node $N_n + M_m + 3$. In the former case, we may replace the arc uv and the subpath from v to $N_n + M_m + 3$ with a single arc connecting u to $N_n + M_m + 3$ without affecting the profit. By repeating similar transformations, if necessary, we may obtain a path that visits at most one node corresponding to the same A_i or B_i.

Therefore, we can obtain an optimal path using only arcs with cost 0. If the path does not visit a node corresponding to a set A_i or B_i, then we can insert such a node by using an arc with zero cost. As a result, the elements of Nonuniform Weighted Max-Rep corresponding to the nodes of the path form a feasible solution of Nonuniform Weighted Max-Rep. Clearly, $a_i^* \in A_i$ and $b_i^* \in B_i$ are in the feasible solution if and only if the corresponding nodes are in the path. From the demand assignment rule in sMTP construction, the sum of the associated weight of edges between a_i^*'s and b_i^*'s is the same as the profit realized along the path. Conversely, it is not difficult to see that any feasible solution of the Nonuniform Weighted Max-Rep, the path visiting the corresponding nodes between nodes 1 and N_n+M_m+3 has the same profit as the sum of the weights achieved by the feasible solution. Therefore, optimal solutions of the Nonuniform Weighted Max-Rep and the constructed sMTP have the same objective value. $\square$

Lemma 3.2.5 implies the inapproximability result of Max-Rep in Kortsarz (2001) is transferred to sMTP.

Theorem 3.2.6. *Even when there is no bound on the capacity, and the costs, revenues, and demands are all binary, sMTP is inapproximable within the factor of* $2^{\log^{1-\varepsilon} n}$ *for any* $\varepsilon > 0$ *unless* $NP \subseteq DTIME(n^{\mathrm{poly}\log n})$.

3.3 Approximation of some special classes of sMTP

In this section, we devise approximation algorithms for two special classes of sMTP, that used as a subroutine of the approximation algorithm for sMTP, in Section . In Subsection 3.3.1, we consider a special class of sMTP's referred to as ssMTP, that have a single supply node which is known to be NP-hard (Kim 2015). We will show ssMTP admits an FPTAS. In Subsection 3.3.2, we consider another class of sMTP's referred to as the *t-separable* sMTP for which we develop an $O(n^{1/2}\log r_{\mathrm{ratio}})$-approximation algorithm, where $r_{\mathrm{ratio}} = \max\{r(k,l)\}/\min\{r(k,l)\}$. Then in Subection 3.3.3, we suggest a method for boosting the approximability from Subsection 3.3.2 for large r_{ratio}. In doing so, the FPTAS for ssMTP in Subsection 3.3.1 is used as a subroutine.

Note that if a path P is fixed in a sMTP, the problem is reduced to a minimum cost flow problem and its optimal x, denoted by x^P, is polynomially computable. Let $c_P(k,l)$ be the logistic cost of the subpath of P from k to l. Then the objective value of sMTP is given as

$$f(P) = \sum_{(k,l)\in D} \big(r(k,l) - c_P(k,l)\big)x^P_{kl}. \tag{3.1}$$

The following lemma will be used for approximation factor analysis of proposed algorithms throughout Section 3.3 and 3.4. Similar lemmas are used for devising approximation algorithm for other problems (Kleinberg 1996, Balcan & Blum 2006, Chekuri et al. 2012).

Lemma 3.3.1. *Suppose for a given sMTP, there are β problems,* sMTP_1, ..., sMTP_β *defined on the same digraph such that each* sMTP_i, $1 \leq i \leq \beta$, *has the objective function f_i satisfying*

- $0 \leq f_i(P) \leq f(P) \leq \sum_{i=1}^{\beta} f_i(P)\ \forall\ P$, *and*
- *there is an optimal path P^* of the given sMTP such that $f_i(P^*) > 0$ for at most γ i's.*

Then there is i such that an α-approximation of sMTP_i *is an $\alpha\gamma$-approximation of the given sMTP.*

Proof: Since $f(P^*) \leq \sum_{i=1}^{\beta} f_i(P^*)$ and at most γ sMTP_i's are positive, there is i' such that $f_{i'}(P^*) \geq f(P^*)/\gamma$. Let $\hat{P}$ be an α-approximation solution of $\text{sMTP}_{i'}$. Then $f(P^*) \leq f_{i'}(P^*)/\gamma \leq f_{i'}(\hat{P})/\alpha\gamma \leq f(\hat{P})/\alpha\gamma$. □

Corollary 3.3.2. *Let D_i, $1 \leq i \leq \beta$, be s-d pair sets on a digraph G such that $D = \bigcup_{i=1}^{\beta} D_i$. Then there is i such that an α-approximation solution of the sMTP, (G, c, D_i, d, p, U), is an $\alpha\beta$-approximation solution of the sMTP, (G, c, D, d, p, U).*

3.3.1 An FPTAS for the single-source sMTP

The single-source sMTP or ssMTP is defined as the sMTP in which $d(k, l) = 0$ for all $k > 1$. We propose an FPTAS for ssMTP based on a pseudo-polynomial dynamic

programming algorithm which is in turn enabled by the acyclicity of the underlying digraph. Since there are objective terms with factors of the form, revenue minus cost, even when we scale the revenues or costs so that the ratio of the minimum to the maximum value is bounded by the $1+\varepsilon$ for a sufficiently small $\varepsilon > 0$, the profitability of a trade between an s-d pair may not be preserved if the ratio of revenue to logistics cost of the pair is less than $1+\varepsilon$. Furthermore, such a factor is multiplied by the traded volume, we need be careful not to let a small profit make a trade profitable in the scaled problem. To address these concerns, we will propose a refined recurrence relation that involves appropriate rounding.

The unique supply node enables us to simplify notation: $d_l := d(1,l)$, $r_l := r(1,l)$, and $x_l := x_{1l}$ for $1 < l \leq n$. Let $c_i^{\max}$ be the maximum distances from node 1 to node i, namely $c_i^{\max} = \sum_{j=1}^{i-1} c_{j(j+1)}$ for $i = 2, \ldots, n$. Let $D(j, \Gamma_j, P)$ be the minimum of total trade $\sum_{k=2}^{j} x_k$ of the subproblem induced by the nodes from 1 to j, whose 1-j path is restricted to have distance no greater than $c_{1j} + \Gamma_j$, and whose profit to be no less than P. Then we get the following relation.

$$D(j, \Gamma_j, P) = \min_{i<j,\ 0 \leq x_j \leq d_j} \Big\{ D\left(i, \Gamma_j - (c_{ij} + c_{1i} - c_{1j}), P - (r_j - c_{1j} - \Gamma_j)x\right) + x_j \Big\}. \tag{3.2}$$

We now scale so that P, Γ_j, and x_i have polynomially many distinct values in (3.2). The maximum profit attainable from trading between single s-d pair is given by $\mathrm{ssMTP}_{\max} = \max_{1 \leq i \leq n} \{(r_i - c_{1i})d_i\}$. The optimal objective value falls onto the interval $[\mathrm{ssMTP}_{\max}, n\mathrm{ssMTP}_{\max}]$. We underestimate the objective values from trades at each node as the closest multiple of $1_{obj} := \mathrm{ssMTP}_{\max}/\lceil n/\varepsilon \rceil$. Then optimal solution of the scaled problem is a $1+\varepsilon$-approximation solution of the original problem.

Let $\Delta c_{\max} = \max_{2 \le i \le n}\{c_i^{\max} - c_{1i}\}$ and C be the set of numbers in the interval $[0, \Delta c_{\max}]$ that can be converted to base $\lceil n/\sqrt{\varepsilon} \rceil$ with exactly 3 significant figures: $C = \{0\} \cup \{\lceil n/\sqrt{\varepsilon} \rceil^k l : 0 \le k \le \log_{n/\sqrt{\varepsilon}} \Delta c_{\max} - 2, 1 \le l \le \lceil n/\sqrt{\varepsilon} \rceil^3 - 1,\ l \in \mathbb{Z}_+\}$.

For $2 \le j \le n$, we round Γ_j to its closest value $\hat{\Gamma}_j$ in $C \cap \{c : 0 \le c \le c_j^{\max} - c_{1j}\ \}$. This may misestimate a cost by at most ε/n^2 times its actual value. Also let $X_j = \{x : 0 \le x \le d_j,\ x = d_j nk/\lceil n^2/\varepsilon \rceil$ for some $k \in \mathbb{Z}_+\}$.

Write as $\hat{D}(j, \hat{\Gamma}_j, \hat{P})$ the minimum total trade $\sum_{k=2}^{j} x_k$ of a solution of the subproblem induced by the nodes from 1 to j whose 1-j path is restricted to have distance no greater than $\hat{\Gamma}_j$, and the profit to be at least $\hat{P}1_{obj}$. If $\hat{P}1_{obj}$ is unattainable, $\hat{D}(j, \hat{\Gamma}_j, \hat{P})$ is set to $+\infty$. Then we get the scaled dynamic relation where $\hat{D}(j, \hat{\Gamma}_j, 0)$ $= 0$ for $1 \le j \le n$.

$$\hat{D}(j, \hat{\Gamma}_j, \hat{P}) = \min_{\hat{P} - \lceil n/\varepsilon \rceil \le \hat{P}_i < \hat{P}, i<j} \Big\{ \hat{D}(i, C(\hat{\Gamma}_j, i, j), \hat{P}_i) + \min\{x : x \in X_j, (r_j - c_{1j} - \hat{\Gamma}_j)x \ge (\hat{P} - \hat{P}_i)1_{obj}\} \Big\}, \tag{3.3}$$

where $C(\hat{\Gamma}_j, i, j)$ is the element of C which, if added by $c_{ij} + c_{1i} - c_{1j}$, is closest to $\hat{\Gamma}_j$. Return $\hat{x}$, as the solution of the scaled problem, which corresponds to the maximum value of a $\hat{P}$ whose $\hat{D}(n, \hat{\Gamma}_n, \hat{P})$ is less than or equal to U. To go through the recurrence (3.3), we need to compute $O(n \times \log \Delta c_{\max}(n/\sqrt{\varepsilon})^3 \times n^2/\varepsilon)$ entries, each requiring $O(n^2/\varepsilon)$ time at maximum. Thus total computation time is $O(n^7 \log \Delta c_{\max}/\varepsilon^3)$, polynomial in the input length and $1/\varepsilon$.

Theorem 3.3.3. *ssMTP admits an FPTAS: an $(1+\varepsilon)$-approximation can be done in $O(n^7 \log \Delta c_{\max}/\varepsilon^3)$ time.*

Proof: We will show the solution $\hat{x}$ is a $1/(1-3\varepsilon)$-approximation solution of ssMTP,

as the difference between the objective value of $\hat{x}$ and the optimal value of ssMTP does not exceed $3\varepsilon\text{ssMTP}_{\max}$.

Suppose the path of $\hat{x}$ visits nodes $v_1(=1)$-v_2-...-v_{k-1}- $v_k(=n)$, trading $\hat{x}_{v_i}$ from node 1 to v_i for $i \geq 2$. Then its scaled objective value is

$$\overline{\text{OBJ}}_{p,c,x}(\hat{x}) = \sum_{i=1}^{k} \left\lfloor \left((r_{v_i} - c_{1v_i})\hat{x}_{v_i} - \hat{\Gamma}_{v_i}\hat{x}_{v_i} \right) / 1_{\text{obj}} \right\rfloor 1_{\text{obj}}. \tag{3.4}$$

If P is treated exactly, namely not scaled in (3.3), we can attain an objective value at least

$$\overline{\text{OBJ}}_{c,x}(\hat{x}) = \sum_{i=1}^{k} ((r_{v_i} - c_{1v_i})\hat{x}_{v_i} - \hat{\Gamma}_{v_i}\hat{x}_{v_i}). \tag{3.5}$$

Since $k \leq n$, the original objective function is no greater than (3.4) by $n1_{\text{obj}} \leq \varepsilon\text{ssMTP}_{\max}$.

If in addition Γ is treated exactly, (3.3) guarantees an objective value no less than

$$\overline{\text{OBJ}}_{x}(\hat{x}) = \sum_{i=1}^{k} ((r_{v_i} - c_{1v_i})\hat{x}_{v_i} - \Gamma_{v_i}\hat{x}_{v_i}). \tag{3.6}$$

Note that $(r_{v_i} - c_{1v_i})\hat{x}_{v_i} - \hat{\Gamma}_{v_i}\hat{x}_{v_i} \geq 0$ for every $2 \leq i \leq k$. For otherwise setting $\hat{x}_{v_i} = 0$ would increase $\overline{\text{OBJ}}_{p,c,x}(x)$. Hence we have $\hat{\Gamma}_{v_i}\hat{x}_{v_i} \leq (r_{v_i} - c_{1v_i})\hat{x}_{v_i} \leq \text{ssMTP}_{\max}$. Since $\hat{\Gamma}_i$ is obtained by at most n additions at each of which its value increases by the factor of ε/n^2, the cumulated cost overestimation is no greater than ε/n times $\hat{\Gamma}_i$. Therefore,

$$|\overline{\text{OBJ}}_{c,x}(\hat{x}) - \overline{\text{OBJ}}_{x}(\hat{x})| \leq \sum_{i=1}^{k} \frac{\varepsilon}{n}\hat{\Gamma}_{v_i}\hat{x}_{v_i} \leq \varepsilon\text{ssMTP}_{\max}.$$

Since the value of x_j can be chosen within the accuracy $d_j\varepsilon/n$, the difference in

objective value due to the scaling of the range of x is bounded by $\sum_{i=1}^{k}(r_{v_i} - c_{1v_i} - \hat{\Gamma}_{v_i})d_{v_i}\varepsilon/n \leq \varepsilon\text{ssMTP}_{\max}$. In sum, the objective value of $\hat{x}$ can not be greater than the optimal objective value of SMTP by more than $3\varepsilon\text{ssMTP}_{\max}$. □

Note that we can also apply the FPTAS to the ssMTP in which the commodity flow is unsplittable by setting $X_j = \{0, d_j\}$. It is extendible for the ssMTP with multiple commodities between the same s-d pair by duplicating each node to nodes with consecutive indices, and setting the cost between the duplication of the same node to zero.

Corollary 3.3.4. *The ssMTP with unsplittable commodity and the ssMTP with multiple commodities per s-d pair admit an FPTAS.*

Consider a sMTP, not necessarily singly sourced, whose demand $d(k, l) > 0$ always satisfies $k \leq t \leq l$. We call such a sMTP, *t-intersecting*. Suppose, in addition, every path is required to stop at the node t and the subpath up to node t is fixed in advance. And we call the problem *t-intersecting sMTP with fixed subpath*. We can transform this class of sMTP to a single-source sMTP with multiple commodities per node by changing the supply node of each s-d pair to node t if its index less than t and compensating for the resulting reduction in logistic costs by subtracting it from the revenue: for each $i < t$ with $d(i, j) > 0$, we create an s-d pair (t, j) with the same demand but with the revenue equal to the original revenue minus the sum of the cost of arcs of the subpath from node i to node t. Therefore we get the following corollary.

Corollary 3.3.5. *The t-intersecting sMTP with a fixed subpath admits an FPTAS.*

Corollary 3.3.4 and 3.3.5 will be used as subroutines of the proposed approxima-

tion algorithms in the following sections.

Remark 3.3.6. If we decompose a given sMTP with $D_i = \{(k,l) \in D | k = i\}$, we get n ssMTP's. Thus, by Corollary 3.3.2, the best of the $O(1)$-approximation of a resulting ssMTP obtained by the FPTAS is an $O(n)$-approximation of the given sMTP.

3.3.2 An approximation algorithm for the t-separable sMTP

Definition 3.3.7. We call a sMTP *t-separable* if there exists a node t satisfying the followings : i) $k \leq t \leq l$ for all $(k,l) \in D$, and ii) $c_{it} = 0$ for $1 \leq i < t$ and $c_{tj} = 0$ for $t \leq j < n$.

Consider an instance of t-separable sMTP denoted by the septuple (G, c, D, d, p, U, t). We decompose it into sMTP's by replacing D with each set D_i of the s-d pairs (k,l) whose $r(k,l)$ have ratios to $p_{\max}$ belonging to the interval $[(1+\varepsilon)^i, (1+\varepsilon)^{i+1})$. Thus there are $O(\log r_{\text{ratio}})$ decomposed problems in each of which there is $\hat{r}$ such that $\hat{r} \leq r(k,l) \leq (1+\varepsilon)\hat{r}$ for all its s-d pairs (k,l). Now we apply Algorithm 3.3.8 to each of the decomposed problems. Recall that we denote our path P only ny the nodes where a positive amount is traded.

Algorithm 3.3.8.

1. *Set $L_0 \leftarrow \{1, \cdots, t-1\}$ and $i \leftarrow 0$.*

2. *On the subgraph induced by node set L_i, find a path P_i(no restrictions on endpoints) whose cost is no greater than $(1+\varepsilon)\hat{r}/3$ and whose number of nodes is no less than $n^{1/2}$.*

3. *If none, go to Step 4. Else, find a* $(1+\varepsilon)$*-approximation solution of the restricted problem whose subpath before* t *is fixed to* P_i*. Set* $L_{i+1} \leftarrow L_i \setminus V(P_i)$*, increase* i *by* 1*, and go to Step 2.*

4. *Let* $S \leftarrow L_i$*. For each case when the path contains exactly one node from* S *before* t*, find a* $(1+\varepsilon)$*-approximation solution, and choose the best of such a solution.*

5. *Return the best of the solutions from Step 3.*

Note the path P_i in Step 2 can be computed by a polynomial time algorithm for computing a shortest path with a predetermined number of nodes (e.g. Cheng & Ansari (2004)). The $(1+\varepsilon)$-approximations in Step 3 and 4 can be done by the FPTAS for ssMTP by Corollary 3.3.5 and 3.3.4, respectively.

Lemma 3.3.9. *Algorithm 3.3.8 guarantees an* $O(n^{1/2})$*-approximation for each problem obtained by the decomposition.*

Proof: Given a sMTP and its subset V' of nodes, Denote by sMTP(V') the restricted problem in which only the s-d pairs (k, l) with $k \in V'$ can be used. We first decompose a given sMTP into sMTP($(\{t\} \cup L_0) \setminus S$) and sMTP($S$). Then it suffices to show that the solutions from Step 3 and 4 are, respectively, $O(n^{1/2})$-approximations of sMTP($(\{t\} \cup L_0) \setminus S$) and sMTP($S$).

Consider first the sMTP(S). The subpath up to t of an optimal path of sMTP(S) should have a logistic cost no greater than $(1+\varepsilon)\hat{r}$. For, otherwise, we can not make any profit from the first node of the path. Also its number of nodes is bound to less than $3n^{1/2} - 2$. Suppose it contains nodes $v_1, \cdots, v_i, t$ with $i \geq 3n^{1/2} - 2$. Then one of three paths having the nodes $\{v_1, \cdots, v_{n^{1/2}}, t\}$, $\{v_{n^{1/2}}, \cdots, v_{2n^{1/2}-1}, t\}$,

or $\{v_{2n^{1/2}-1}, \cdots, v_i, t\}$, should have more than $n^{1/2}$ nodes and its cost should be less than or equal to $(1+\varepsilon)\hat{r}/3$ before t, a contradiction. Therefore, by Lemma 3.3.1 applied to the case $\text{sMTP}_i = \text{sMTP}(\{i\})$ for $i \in S$ and $\gamma = 3n^{1/2} - 2$, an $(1+\varepsilon)$-approximation solution in Step 4 provides an $O(n^{1/2})$-approximation of $\text{sMTP}(S)$.

Now consider the $\text{sMTP}((\{t\} \cup L_0) \setminus S)$. Algorithm 3.3.8 adds a nonempty set to current partition of $L_0 = \{1, \cdots, t-1\}$ in nested manner whenever it finds a path P_i satisfying the condition of Step 2. Since the path has at least $n^{1/2}$ nodes, the partition may not have more than $n^{1/2}$ sets. By Corollary 3.3.2, an optimal solution of the restricted sMTP in which the path may use only the nodes from $V(P_i)$ before t is an $O(n^{1/2})$-approximation solution of $\text{sMTP}((\{t\} \cup L_0) \setminus S)$. Now we will see an optimal solution of the restricted problem whose subpath before t is fixed to P_i is an $O(1)$-approximation solution of $\text{sMTP}(\{t\} \cup V(P_i))$. If so, the best of the solutions found in Step 3 should be an $O(n^{1/2})$-approximation of $\text{sMTP}((\{t\} \cup L_0) \setminus S)$.

Suppose an optimal solution x^* of $\text{sMTP}(\{t\} \cup V(P_i))$ uses a path $l_1 - l_2 - \cdots - l_a - t - r_1 - r_2 - \cdots - r_b$. Since a profitable trade should have the cost of the subpath from t to r_b no greater than $(1+\varepsilon)\hat{r}$, there exist b_1 and b_2 with $1 \le b_1 \le b_2 \le b$ such that each cost of the subpaths from r_1 to r_{b_1}, from r_{b_1} to r_{b_2}, and from r_{b_2} to r_b is no greater than $(1+\varepsilon)\hat{r}/3$. Consider three solutions that have the common subpath before t as x^* but thereafter the subpaths from r_1 to r_{b_1}, from r_{b_1} to r_{b_2}, and from r_{b_2} to r_b, respectively. Let their trades at each node of the paths be the same as x^*. Then the sum of their revenues is the same as the revenue of x^* and each of their logistic costs is no greater than that of x^*. Thus Corollary 3.3.2 implies that the best of the three feasible solutions is a 3-approximation of x^*. Let the solution be x'.

The condition on P_i implies the cost from l_1 to t is no greater than $(1+\varepsilon)\hat{r}/3$. Therefore the logistic cost either up to t or after t can not exceed $(1+\varepsilon)\hat{r}/3$. Hence for every (k,l) with $k < t \leq l$, $r(k,l)$ minus the logistic cost from k to l falls onto the interval $[(1-2\varepsilon)\hat{r}/3,\ (1+\varepsilon)\hat{r}]$. Thus the profit from a unit trade between an s-d pair,which varies depending on the path, does not differ by more than $3(1+\varepsilon)/(1-2\varepsilon)$. This means $(1+\varepsilon)$-approximation of the restricted problem whose subpath before t is fixed to P_i has the objective value no lesser than $3(1+\varepsilon)^2/(1-2\varepsilon)$ times the objective value of x'. In sum, the path is $O(1)$-approximation solution of the restricted sMTP whose path may use only the nodes from $V(P_i)$ before t. □

Theorem 3.3.10. *A t-separable sMTP is approximable within $O(n^{1/2}\log r_{\text{ratio}})$ times its optimum.*

Proof: The decomposition of t-separable sMTP compromised the approximation guarantee by the factor $O(\log r_{\text{ratio}})$. And by Lemma 3.3.9, each resulting sMTP is $O(n^{1/2})$-approximable. Combining the two, we can approximate the t-separable sMTP within $O(n^{1/2}\ \log r_{\text{ratio}})$ times its optimum. □

The reduction in the proof of Lemma 3.2.5 implies that the Nonuniform Weighted Max-Rep is a t-separable sMTP with $r_{\text{ratio}} = 1$. Hence the approximation algorithm for the t-separable sMTP from this subsection, applied to the Max-Rep, guarantees approximation factor, $O(n^{1/2})$, the same order as the current best by Peleg (2007).

3.3.3 Boosting approximability of the t-separable sMTP's with large r_{ratio}

The factor $\log r_{\text{ratio}}$ of the approximation guarantee $O(n^{1/2}\log r_{\text{ratio}})$ from Subsection 3.3.2 is due to the partition bounding the ratio between any pair of revenues of

each problem to $1+\varepsilon$. The value $\log r_{\text{ratio}}$ can be as large as $\Omega(n)$. This subsection boosts approximability of the problem for large r_{ratio}'s by proposing an intermediate and coarser partition which combined with the t-separability results in the following lemma.

Lemma 3.3.11. *Let a constant $\alpha > nm/\varepsilon$. If the t-separable sMTP with $r_{\text{ratio}} \leq \alpha$ is approximable within $f(n,m,\alpha)$ times its optimum, then the t-separable sMTP with $r_{\text{ratio}} > \alpha$ is approximable within $O(\max\{f(n,m,\alpha), n^{1/2}(\log r_{\text{ratio}}/\log\alpha)^{1/2}\})$ times its optimum.*

Proof: Assume we are given an algorithm approximating the t-separable sMTP with $r_{\text{ratio}} \leq \alpha$ with the factor $f(n,m,\alpha)$. We will denote a given t-separable sMTP with $r_{\text{ratio}} > \alpha$ by the septuple (G,c,D,d,p,U,t). We sort the s-d pairs, $(k,l) \in D$, according to which interval $[\alpha^q, \alpha^{q+1})$ its $r(k,l)$ belongs. We then decompose the t-separable problem into two problems by partitioning q's to their parity.

We first consider the problem with the s-d pairs having $r(k,l)$'s in the intervals of odd q's. The other problem determined by the even q's can be treated analogously. Let D_r be the set of s-d pairs whose $r(k,l)$ fall into $[\alpha^{2r+1}, \alpha^{2r+2})$. If two $r(k,l)$'s belong to different intervals, the larger is at least $nm/\varepsilon < \alpha$ times the smaller. Let M_1 and M_2, respectively, be the minimum and the maximum values of r whose D_r is nonempty. Then $M := M_2 - M_1$, an upper bound of the number of nonempty D_r, is $O(\log r_{\text{ratio}}/\log\alpha)$.

In an approximation analysis, an arc ij with cost $c_{ij} \leq r(k,l)/\alpha \leq r(k,l)/(nm/\varepsilon)$ is insignificant to the s-d pair (k,l): since a path has at most n arcs and covers at most m s-d pairs, the total logistic cost induced from the such arcs is at most $\varepsilon \max\{r(k,l)d(k,l)\}$ which is, in turn, less than ε times optimal value. It contributes

at most $O(1)$ factor in approximation guarantee for the problem. On the other hand, an arc with a cost greater than $r(k,l)$ will not be used for a trade between s-d pair (k,l). In these senses, we call an arc ij *significant* to the s-d pairs (k,l) of D_r with $k \leq i < j \leq l$ if $c_{ij} \in J_r := [\alpha^{2r}, \alpha^{2r+2})$, $M_1 \leq r \leq M_2$, *insignificant*, otherwise. Note that if $c_{ij} \in J_{M_1-1} := [0, \alpha^{2M_1})$, ij is insignificant to every s-d pair.

Note that each of the s-d pairs served by a path P belongs to one of three groups (not necessarily mutually exclusive), the s-d pairs using a significant arc both before and after t in P, the s-d pairs using only insignificant arcs before t, and the s-d pairs using only insignificant arcs after t. Thus if we decompose a t-separable sMTP into three sMTP's accordingly, namely, by restricting the choice of path to include or exclude a significant arc before or after node t, they also satisfy the conditions in Corollary 3.3.2. Hence the best solution from the restricted problems is a 3-approximation of the original t-separable sMTP. We now show how to approximate each problem.

Case 1: Significant arcs before and after t

We first consider the t-separable sMTP in which every trade should use a significant arc both before and after node t. For each r satisfying $M_1 \leq r \leq M_2+1$, let LT_rLH_r and RT_rRH_r be the arcs of P such that 1) their cost is greater than α^{2r}, 2) $LH_r \leq t \leq RT_r$, and 3) they are closest to node t in P, i.e. an end node of them is closest to t as a node of P than an end node of any other arc satisfying 1) and 2). We will denote by LT_r and LH_r, or RT_r and RH_r the nodes 1, or n, respectively, if such an arc does not exist.

Let us make some observations. First, the node indices LT_r and LH_r decrease

in r while RT_s and RH_s increase in s. Note that the node sets $\{LH_{r+1}, \ldots, LT_r\}$ and $\{RH_s, \ldots, RT_{s+1}\}$ are mutually disjoint for every $r, s \in [M_1, M_2 + 1]$. Second, should a trade be profitable between s-d pair (k, l) with $r(k, l) \in [\alpha^{2r+1}, \alpha^{2r+2})$, it ought to be $LH_{r+1} \leq k \leq LT_r$ and $RH_r \leq l \leq RT_{r+1}$. If $k < LH_{r+1}$ or $RT_{r+1} < l$, the logistic cost from k to l exceeds the revenue. If $LT_r < k$ or $l < RH_r$, a trading between (k, l) will not use any significant arc.

The observations enable the following dynamic programming algorithm. For each sextuple (a, b, c, d, e, f) of nodes with $1 \leq a \leq b \leq c \leq t \leq d \leq e \leq f \leq n$, let $F_r(a, b, c, d, e, f)$ be the maximum profit obtainable from the s-d pairs (k, l) satisfying $k \in \{a, a+1, \ldots, b\}$, $l \in \{e, e+1, \ldots, f\}$, and $(k, l) \in D_r$ where $(LH_{r+1}, LT_r, LH_r, RT_r, RH_r, RT_{r+1}) = (a, b, c, d, e, f)$. Note that all the arcs ij with $LH_r \leq i < j \leq RT_r$ are insignificant to any $(k, l) \in D_r$.

Therefore, we can approximate $F_r(a, b, c, d, e, f)$ within an $O(1)$ error in the approximation factor by solving the sMTP with the following restrictions: D is restricted to s-d pairs (k, l) satisfying $k \in \{a, a+1, \ldots, b\}$ and $l \in \{e, e+1, \ldots, f\}$; the path P must use arcs (b, c), (c, t), (t, d), and (d, e); and the arcs between node a and b, and c and d should have a cost less than α^{2r+2}. The r_{ratio} is less than α in this restricted sMTP hence to which can be applied the given approximation algorithm guaranteeing the factor $f(n, m, \alpha)$.

Denote by $G_r(a, f)$ the optimal objective value of the sMTP obtained by replacing D of the sMTP in Case 1 with the union of D_s with $s \leq r$ when $a = LH_{r+1}$ and $f = RT_{r+1}$. Then the optimal value of the sMTP in Case 1 is $G_{M_2}(1, n)$ from the

following relation for $M_1 \leq r \leq M_2$ and $1 \leq a \leq t \leq f \leq n$:

$$G_r(a, f) = \max_{a \leq b < c \leq t \leq d < e \leq f} \{F_r(a, b, c, d, e, f) + G_{r-1}(c, d), G_{r-1}(a, f)\}, \tag{3.7}$$

where $G_r(t, t) = 0$ and $G_{M_1-1}(a, f) = 0$. From the $O(1)$ error in computing F, we can show $G_{M_2}(1, n)$ can be approximated within $O(f(n, m, \alpha))$ times its exact value from (3.7).

Case 2: No significant arc before t

Let sMTP_{ns} be the sMTP restricted to exclude significant arcs before t in its path. We will introduce an $O(n^{1/2}M^{1/2}) = O(n^{1/2}(\log r_{\text{ratio}}/\log \alpha)^{1/2})$-approximation algorithm for sMTP_{ns}. The third case in which significant arcs are excluded after node t is also approximable with the same factor if treated symmetrically.

Again sMTP_{ns} will be denote by the septuple (G, c, D, d, p, U, t). We first transform sMTP_{ns} into an equivalent unrestricted problem, sMTP' so that the solutions of two problems correspond one-to-one with the same objective values. Construct the digraph G' of sMTP' as follows.

- The node set of G' is $L' \cup R$, where $L' := \{1, \cdots, t-1\} \times \{M_1 - 1, M_1, \ldots, M_2\}$ and $R := \{t, \cdots, n\}$. Thus a node of G' is of the form (i, r) or j. Then we call i and j the *first components* of the nodes.

- The arc set of G' is $A_L \cup A_t \cup A_R$, where A_L is the set of arcs connecting $(i, \max\{r, s\})$ to (j, s) for each $ij \in A$ such that $i < j \leq t$ and $c_{ij} \in J_r$, A_t the set of arcs connecting $(i, M_1 - 1)$ to t for each $i \leq t-1$, and A_R the set of arcs connecting i to j for each $ij \in A$ with $t \leq i < j$.

- The cost of each arc of G' is the same as the arc of G which has as the end nodes the first components of its end points.
- For each $r(k,l) \in \bigcup_{i=u+1}^{M_2} D_i$ with $M_1 - 1 \leq u \leq M_2 - 1$ the s-d pair $((k,u),l)$ of G' has the same demand and revenue as the s-d pair (k,l) of G. Each s-d pair (t,l) of G' has the same demand and revenue as the s-d pair (t,l) of G.

Notice that for each node $(j,s) \in L'$, there is a unique arc of G' connecting a node whose first component is i with $i < j$ to (j,s). Therefore to each path of G, $P := l_1 - l_2 - \cdots - l_p - t - r_1 - r_2 - \cdots - r_q$, corresponds a unique path P' of G' whose subpath up to t has the nodes represented by the ordered pairs with first components, $l_1, \cdots, l_p$, which has the subpath $r_1 - r_2 - \cdots - r_q$ right after t and visits the node $(l_p, M_1 - 1)$. Recall that $c_{l_p t} \in J_{M_1-1}$. Also if $c_{l_i l_{i+1}} \in J_u$, P' contains the node (l_{i+1}, s), and its most expensive arc after l_{i+1} has the cost belonging to J_s, then P' contains node $(l_i, \max\{u,s\})$ and the largest cost of an arc after l_i belongs to $J_{\max\{u,s\}}$. Thus, inductively, if P' visits node (l_i, u), its largest cost of an arc of the subpath from l_i to t belongs to J_u.

By construction, the corresponding paths from G and G' have the same cost. Also we will see that in an optimal solution, the corresponding paths have the same revenue from their s-d pairs corresponding to each other. Let x be a feasible solution of sMTP_{ns}, P its path, and P' the corresponding path of sMTP'. If there is a trade between s-d pair (l_i, r_j) in x of sMTP_{ns}, and $r(l_i, r_j) \in D_s$, then due to the restriction of this case, the costs of arcs of l_i-t subpath of P fall onto $\bigcup_{k=M_1-1}^{s-1} J_k$. If P' contains node (l_i, u), the largest cost of an arc of its l_i-t subpath belongs to J_u. Thus P''s containing node (l_i, u) implies that each s-d pair (l_i, r_j) of G is connected by a subpath of P free from a significant arc before t if and only if $r(l_i, r_j) \in \bigcup_{i=u+1}^{M} D_i$.

Therefore, by construction, there is one-to-one correspondence between such s-d pairs (l_i, r_j) of G and the s-d pairs $((l_i, u), r_j)$ of sMTP$'$.

We have shown the equivalence between sMTP$'$ and sMTP$_{\text{ns}}$. Also given an optimal solution of one problem, we can construct an optimal solution of the other: from an optimal path of sMTP$'$, the corresponding path of sMTP$_{\text{ns}}$ can be constructed simply by taking its first components. Conversely, suppose we are given an optimal path P of sMTP$_{\text{ns}}$. Then for each node l of P before t, we replace l with the node of G' with the ordered pair (l, s) where J_s is the interval to which the largest cost of an arc on the l-t subpath of P belongs. Therefore we can apply Algorithm 3.3.12 to sMTP$'$ to obtain a feasible solution of sMTP$_{\text{ns}}$ guaranteed the same approximation factor.

Algorithm 3.3.12.

1. *Set $L'_0 \leftarrow L'$ and $i \leftarrow 0$.*

2. *Find a path P_i in L'_i whose number of nodes is no less than $(nM)^{1/2}$.*

3. *If none, go to Step 4. Else, set $L'_{i+1} \leftarrow L'_i \setminus V(P_i)$. Find a $(1+\varepsilon)$-approximation solution x^i of the* sMTP$'$ *whose path uses exactly the nodes of $V(P_i)$ before t. Set $i \leftarrow i+1$ and go to Step 2.*

4. *Let $S \leftarrow L'_i$. For each case when a path contains exactly one node from S, find a $(1+\varepsilon)$-approximation solution and choose the best of such a solution.*

5. *Return the best of the solutions from Step 3 and 4. Set $N \leftarrow i$.*

As for Algorithm 3.3.8, the computation of path in Step 2 and the approximation

in Step 3 and 4 can be performed in polynomial time.

We now claim that the solution returned by Algorithm 3.3.12 is an $O((nM)^{1/2})$-approximation of sMTP$'$. The argument is similar with but simpler than the proof of Lemma 3.3.9. Denote by sMTP$'(V')$ the sMTP obtained from sMTP$'$ by restricting the s-d pairs to the ones whose origin nodes belong to V'. Decompose sMTP$'$ into sMTP$'((\{t\} \cup L'_0) \setminus S)$ and sMTP$'(S)$. Then it suffices to show that the solutions from Step 3 and 4, respectively, are $O((nM)^{1/2})$-approximations of sMTP$'((\{t\} \cup L'_0) \setminus S)$ and sMTP$'(S)$.

Regarding sMTP$'(S)$, since its every path has less than $(nM)^{1/2}$ nodes from S, by Lemma 3.3.1 with sMTP$_i$ = sMTP$'(\{i\})$ $\forall i \in S$ and $\gamma = (nM)^{1/2}$, one of $(1+\varepsilon)$-approximation solutions from Step 4 provides an $O((nM)^{1/2})$-approximation of sMTP$'(S)$. Consider now sMTP$'((\{t\} \cup L'_0) \setminus S)$. As i increases, at least $(nM)^{1/2}$ elements are excluded from V'. Since $|L'| = O(nM)$, we have $N = O((nM)^{1/2})$. By Corollary 3.3.2, one of $O(1)$-approximation solutions of sMTP$'(\{t\} \cup V(P_i))$ for $0 \leq i < N$ is an $O((nM)^{1/2})$-approximation of sMTP$'((\{t\} \cup L'_0) \setminus S)$. Since all the arcs whose end nodes have indices less than t are insignificant to any s-d pair, fixing the nodes of path to $V(P_i)$ before t does not compromise approximation factor by more than $O(1)$. Therefore, one of $(1+\varepsilon)$-approximation solutions in Step 3 provides an $O((nM)^{1/2})$-approximation of sMTP$'((\{t\} \cup L'_0) \setminus S)$.

Since the worst of the approximation factors from Case 1 and Case 2 is $O(\max\{f(n, m, \alpha), n^{1/2}(\log r_{\text{ratio}}/\log \alpha)^{1/2}\})$, Lemma 3.3.11 follows. □

Theorem 3.3.13. *For any given constant $k \in \mathbb{Z}_+$, the t-separable sMTP admits the following approximation algorithm.*

- $O(n^{1/2} \log r_{\text{ratio}})$*-approximation algorithm if* $\log r_{\text{ratio}} = o(\log m)$,

- $O(n^{1/2}\log m)$-*approximation algorithm if* $\log\log r_{\text{ratio}} < (2k{+}1)\log\log(mn/\varepsilon)$, *and*

- $O(n^{1/2}\log^{1/k} r_{\text{ratio}})$-*approximation algorithm if* $\log\log r_{\text{ratio}} \geq (2k{+}1)\log\log(mn/\varepsilon)$.

Proof: The first case follows from Theorem 3.3.10. Regarding the last two cases, let $\bar{r}_{\text{ratio}}$ be the value of r_{ratio} of given t-separable sMTP.

We now prove the second case by induction on k. Suppose $\log\log\bar{r}_{\text{ratio}} < \log\log(mn/\varepsilon)$. Then $O(n^{1/2}\log\bar{r}_{\text{ratio}})$-approximation algorithm from Theorem 3.3.10 is an $O(n^{1/2}\log m)$-approximation algorithm. Suppose now $(2k-1)\log\log(mn/\varepsilon) \leq \log\log\bar{r}_{\text{ratio}} < (2k+1)\log\log(mn/\varepsilon))$. Then by the induction hypothesis and Lemma 3.3.11 with $r_{\text{ratio}} = \bar{r}_{\text{ratio}}$ and α satisfying $\log\log\alpha = (2k-1)\log\log(mn/\varepsilon)$, we get an approximation factor, $O(\max\{O(n^{1/2}\log m),\ n^{1/2}(\log\bar{r}_{\text{ratio}}/\log\alpha)^{1/2}\}$ which is $O(n^{1/2}\log m)$. The inductive arguments shows that we can approximate the t-separable sMTP within the factor $O(n^{1/2}\log m)$ by applying Lemma 3.3.11 at most k times.

Assume $\log\log\bar{r}_{\text{ratio}} > (2k+1)\log\log(mn/\varepsilon)$ as in the last case. By Theorem 3.3.10, if we take $\log\log r_{\text{ratio}} = \log\log(\bar{r}_{\text{ratio}})/(2k+1)$, we can approximate a t-separable sMTP within the approximation factor of $O(n^{1/2}\log^{1/(2k+1)}\bar{r}_{\text{ratio}})$. Furthermore if we take r_{ratio} and α so that $r_{\text{ratio}} = \log\log\bar{r}_{\text{ratio}}$ and $(2k+1)\log\log\alpha = (2k-1)\log\log\bar{r}_{\text{ratio}}$, by Lemma 3.3.11 and the induction hypothesis, we can approximate the t-separable sMTP within the factor $O(n^{1/2}\log^{1/(2k+1)}\bar{r}_{\text{ratio}})$. Hence by applying Lemma 3.3.11 at most k times, we can get an $O(n^{1/2}\log^{1/(2k+1)}\bar{r}_{\text{ratio}})$-approximation algorithm. □

Remark 3.3.14. Each application of Lemma 3.3.11 multiplies the approximation factor by $O(1)$. Thus if we allow the computation time to be quasi-polynomial, i.e.

$k = O(\sqrt{\log r_{\text{ratio}}})$ we obtain the approximation guarantee $O(n^{1/2}c^{\sqrt{\log r_{\text{ratio}}}})$ for a constant c.

3.4 Approximation of general sMTP

In this section, we establish an approximability of sMTP by reducing sMTP to the t-separable sMTP at the expense of some approximation factor. In so doing, we use another intermediate problem between the two problems, t-intersecting sMTP, which is, recall that, defined as the sMTP's having a node t such that $k \leq t \leq l$ for all $(k, l) \in D$, the first condition of the t-separable sMTP.

3.4.1 Reducing sMTP to t-intersecting sMTP

We will assume by removing any node not in an s-d pair, if necessary, that every node of a given sMTP is an end node of an s-d pair. Also by introducing at most $n - 2$ dummy nodes with an unprofitable demand, we can assume there is $q \in \mathbb{Z}_+$ such that $n = 2^q$.

We first decompose a given sMTP into $O(\log n)$ instances by adopting the idea in Balcan & Blum (2006). Let $D_i := \big\{(k, l) : \exists b \in \mathbb{N} \text{ such that } k \leq n(2b-1)/2^i \leq l\big\} \setminus \bigcup_{j=1}^{i-1} D_j$. Note that the problems, sMTP^i, each obtained by replacing D by D_i in the given sMTP, satisfy the condition of Corollary 3.3.2. There are $\log_2 n$ nonempty D_i's, $D = \bigcup_{i=1}^{\log_2 n} D_i$. And for each $(k, l) \in D_i$, there is b: $1 \leq b \leq 2^{i-1}$ such that

$$n(b-1)/2^{i-1} < k \leq n(2b-1)/2^i < l < nb/2^{i-1}. \tag{3.8}$$

Since the sets of consecutive nodes $\{n(b-1)/2^{i-1}+1, n(b-1)/2^{i-1}+2, \ldots nb/2^{i-1} -$

1} are pairwise disjoint for distinct b's, the sets of s-d pairs satisfying (3.8) are a partition of D_i. Let each of the set be D_i^b, $1 \leq b \leq 2^{i-1}$, and denote by sMTP_b^i the sMTP whose s-d pairs are restricted to each set D_i^b. Given feasible solutions of sMTP_b^i, we can construct a feasible solution of sMTP^i by concatenating the subpaths of sMTP_b^i's corresponding to the node intervals $\{n(b-1)/2^{i-1}+1, n(b-1)/2^{i-1}+2, \ldots nb/2^{i-1}-1\}$. Therefore we can guarantee sMTP^i at least the worst approximation factor guaranteed for a sMTP_b^i. Note each sMTP_b^i is a t-intersecting sMTP with $t = n(2b-1)/2^i$. Thus solving sMTP^i has been reduced to solving t-intersecting sMTP's, sMTP_b^i, $1 \leq b \leq 2^{i-1}$.

Lemma 3.4.1. *Suppose the t-intersecting sMTP admits an $f(n,m,\min\{p_{\max},md_{\text{ratio}}\})$-approximation algorithm, where f is a non-decreasing function. Then sMTP admits an $O(f(n,m, \min\{p_{\max}, md_{\text{ratio}}\})\log n)$-approximation algorithm.*

3.4.2 Reducing t-intersecting sMTP to t-separable sMTP

Finally we reduce the t-intersecting sMTP to the t-separable sMTP. We represent a given t-intersecting sMTP by septuple (G, c, D, d, p, U, t). Suppose we know a priori the last node $\tilde{t}$ such that $\tilde{t} \leq t$, an optimal path visits. We then can transform the sMTP into an equivalent instance in which every optimal path stops by node t. Assume $\tilde{t} \neq t$. Then we may delete arcs kl such that $k < \tilde{t} < l$ and s-d pair (k, l) such that $\tilde{t} < k \leq t$. Then we get a $\tilde{t}$-intersecting sMTP to which we apply the following procedure to get a t-separable sMTP.

Algorithm 3.4.2. ***Process***

1. *For each node i, let $\tilde{c}_{i\tilde{t}} = 0$ if $i \leq \tilde{t}$, and let $\tilde{c}_{\tilde{t}i} = 0$ if $\tilde{t} \leq i$.*
2. *For each s-d pair (k, j) such that $1 \leq k \leq \tilde{t} \leq t \leq l \leq n$,*

   let $\tilde{r}(k, l) \leftarrow r(k, l) - c_{k\tilde{t}} - c_{\tilde{t}l}$.
3. *For each node pair i and j such that $1 \leq i < j \leq \tilde{t}$, let $\tilde{c}_{ij} = c_{ij} + c_{j\tilde{t}} - c_{i\tilde{t}}$.*
4. *For each node pair i and j such that $\tilde{t} \leq i < j \leq n$, let $\tilde{c}_{ij} = c_{ij} + c_{\tilde{t}i} - c_{\tilde{t}j}$.*

Clearly, the procedure reduces both the cost of subpath containing $\tilde{t}$ from i to j and the revenue $r(i, j)$ by $c_{i\tilde{t}} + c_{\tilde{t}j}$ and hence does not change the objective value while making the costs of arcs having $\tilde{t}$ as an end node to vanish to 0. For each $1 \leq \tilde{t} \leq t$, denote the obtained $\tilde{t}$-separable sMTP by $\text{sMTP}_{\tilde{t}}$. Note that there is $\tilde{t}$ such that an optimal solution of $\text{sMTP}_{\tilde{t}}$ is an optimal solution of the given t-intersecting sMTP.

Consider $\text{sMTP}_{\tilde{t}}$ for any $\tilde{t}$ and write its s-d pairs, demands, revenues and costs by $\tilde{D}$, $\tilde{d}$, $\tilde{r}$ and $\tilde{c}$. We will scale $\text{sMTP}_{\tilde{t}}$ to make the ratio of minimum to maximum value of $\tilde{r}(k, l)\tilde{d}(k, l)$ no greater than m/ε. From $\tilde{D}$, discard (k, l)'s such that $\tilde{r}(k, l)d(k, l) \leq \max_{(k,l)\in\tilde{D}} \tilde{r}(k, l)d(k, l)\varepsilon/m$. Since $\max_{(k,l)\in\tilde{D}} \tilde{r}(k, l)d(k, l)$ is a lower bound on the optimal value, it does not make the optimal value of the resulting problem different from the original one more than ε times the original optimum. This scaling bounds the ratio of the minimum to the maximum value of $\tilde{r}(k, l)d(k, l)$ no greater than m/ε as desired.

Note that the values of n, m, and d of $\text{sMTP}_{\tilde{t}}$ are the same as the original t-intersecting sMTP. Regarding $\tilde{r}_{\text{ratio}}$ we have the following lemma.

Lemma 3.4.3. *The value,* $\min\{p_{\max}, md_{\text{ratio}}/\varepsilon\}$ *of t-intersecting sMTP is no less than $\tilde{r}_{\text{ratio}}$ of* $\text{sMTP}_{\tilde{t}}$.

Proof: Due to integrality of parameters, we have $\tilde{r}_{\text{ratio}} \leq \tilde{p}_{\max}$ in sMTP$_{\tilde{t}}$. Furthermore, since we have $\tilde{r}(k,l) = r(k,l) - c_{k\tilde{t}} - c_{\tilde{t}l} \leq r(k,l) - c_{kl}$, $\tilde{p}_{\max}$ of sMTP$_{\tilde{t}}$ is no greater than $p_{\max}$ of the t-intersecting sMTP. On the other hand, due to the additional condition that the ratio of minimum to maximum value of $\tilde{r}(k,l)\tilde{d}(k,l)$ is no greater than m/ε, r_{ratio} of sMTP$_{\tilde{t}}$ is no greater than $md_{\text{ratio}}/\varepsilon$. □

Then we get the following lemma.

Lemma 3.4.4. *From an $f(n,m,r_{\text{ratio}})$-approximation algorithm for the t-separable sMTP, we can derive an $O(f(n,m,\min\{p_{\max},md_{\text{ratio}}/\varepsilon\}))$-approximation algorithm for the t-intersecting sMTP. Here, f is a monotone non-decreasing function.*

Combining Lemma 3.4.1 and 3.4.4, we have the following lemma.

Lemma 3.4.5. *From an $f(n,m,r_{\text{ratio}})$-approximation algorithm for the t-separable sMTP, we can derive an $O(f(n,m,\ \min\{p_{\max},md_{\text{ratio}}/\varepsilon\})\ \log n)$-approximation algorithm for sMTP. Here f is a monotone non-decreasing function.*

Theorem 3.4.6. *Let $\delta > 0$ be an arbitrary constant. The t-intersection sMTP is approximable within the factor, $O\Big(n^{1/2} \min\Big\{ \log p_{\max},\ \max\{ \log m,\ \log^{\delta} \min\{p_{\max}, d_{\text{ratio}}\}\}\Big\}\Big)$. The general sMTP is approximable within the factor, $O\Big(n^{1/2} \log n \min\Big\{ \log p_{\max},$ $\max\{ \log m,\ \log^{\delta} \min\{p_{\max}, d_{\text{ratio}}\}\}\Big\}\Big)$.*

Proof: Note the cases in Theorem 3.3.13 exhaust every sMTP, and their approximation factors can be unified as $O\Big(n^{1/2} \min\Big\{ \log r_{\text{ratio}},\ \max\{ \log m,\ \log^{\delta} r_{\text{ratio}}\}\Big\}\Big)$. Applying Lemma 3.4.4 and 3.4.5, respectively, to this factor, we get the approximation guarantees for the t-intersecting sMTP and the general sMTP as in the theorem.

□

3.5 Applying to Other Problems

sMTP assumes the logistic cost is linear in the load level. Therefore the model is appropriate when the vehicle operation cost is a fixed cost. If there is an additional cost proportional to total travel distance, from the reasons such as the total cost of fuel is proportional to total weight that include the vehicle's weight, or in the case that the driver's labor cost is arranged as total travel time. We can transform the problem into an equivalent sMTP by adding a profitable s-d pair from node 1 to n. The addition of profit changes the hardness of the problem, in contrary to the case that aims to get an optimal solution. If we consider the sMTP with additional cost directly, we can show the sMTP is not approximable within any factor.

Theorem 3.5.1. *sMTP with additional cost linear in total travel distance cannot be approximate within any factor.*

proof: Let the additional cost be the value obtained by multiplying the driving distance by a constant k. We will show the inapproximability of this problem by slightly modifying the reduction in Lemma 3.2.5. Specifically, we will reduce from the decision version of Max-Rep, which asks whether there exists a feasible solution with objective function value greater than α for a given instance of Max-Rep.

First, we increase the cost of all edges from vertices with indices less than or equal to $N_n + 2$ to vertices with indices greater than t by α/k, and increase the revenue obtained from all existing unit transactions by the same amount. This decreases the objective function value by α without changing the feasible solution set (the objective function value of the original sMTP remains unchanged, while only the vehicle operation cost increases). In other words, the problem of finding the optimal

objective function value of this problem is equivalent to the question of whether the given Max-Rep instance has a "yes" answer. Therefore, assuming the existence of a polynomial-time approximation algorithm with a positive coefficient for this problem implies that we can solve the decision version of Max-Rep, which in turn implies $P = NP$. $\square$

The validity of the approximation relies on Corollary 3.3.2, on the decomposition of s-d pairs into the problems with the same feasible paths as the original problem but whose objective function is no less than the original one, and the availability of a good approximation algorithm for the t-intersecting sMTP with a fixed subpath, e.g. an FPTAS from Corollary 3.3.5. Here is a list of the variants of sMTP satisfying these two conditions that are, hence, approximable within the same order of the factor as sMTP.

- sMTP in which the revenue is subadditive in trade volume,
- sMTP whose logistic cost is supperadditive in the total load carried along at each arc,
- sMTP with unsplittable commodities, and
- sMTP in which total travel distance is bounded.

In Subsection 3.5.1, we aim to present the first approximation algorithm for the Traveling Repairman Problem with Profits (TRPP), which is one of the most representative cases, in addition to the itemized cases mentioned above. In Subsection 3.5.2, we prove that TRPP is hard to approximate within a constant.

3.5.1 Approximation algorithm for TRPP

Problem 3.5.2. Traveling Repairman Problem with Profit, TRPP Let $G = (V, E)$ be a complete undirected graph with $V = \{1, \cdots, n\}$. A revenue r_i is obtained from visiting each node $i \in V$ except for node 1. It takes time d_{ij} to travel across each edge $ij \in E$. If a repairman arrives at node i at time t_i, he obtains a profit of $r_i - t_i$. TRPP is the problem of finding a path for the repairman starting from node 1 at time 0, which maximizes the total profit. We assume that the triangle inequality is satisfied by the edge weights d_{ij}.

Since d_{ij}'s satisfy triangle inequality, it suffices to consider only paths where a positive profit is realized at every node. In this section, we present an algorithm for approximating the TRPP. To accomplish this, we first introduce an intermediate problem called TRPP$_\alpha$. We then use an algorithm based on the previously proposed approximation algorithm for the orienteering problem to approximate TRPP$_\alpha$.

Problem 3.5.3. Orienteering problem Consider a complete undirected graph $G = (V, E)$ with $V = 1, \cdots, n$. Each node $i \in V$ has a profit of p_i which can be obtained by visiting it. The time required to cross an edge $ij \in E$ is d_{ij}. Given a time limit $L > 0$, the goal is to find a 1-n path with the sum of edge times bounded by L, while maximizing the sum of node profits.

Let $p_i = r_i - d_{1i}$ be the *potential profit* of node $i \in V$, which represents the maximum profit that can be obtained from node i.

Problem 3.5.4. TRPP$_\alpha$ As TRPP$_\alpha$ we refer to a TRPP where the ratio of the minimum to maximum potential profits of nodes is less than a constant $\alpha > 0$. In other words, there exists a value $p > 0$ such that for all $i \in V \backslash 1$, we have $p_i \in [p, \alpha p)$.

Note that Theorem 3.5.10 implies that there is no polynomial time approximation scheme for TRPP_α, for any $\alpha > 0$. We adopt the notation of Blum et al. (2007) and use $t_i(P)$ to denote the arrival time at node i via a path P. We also define $e_i(P) = t_i(P) - d_{1i}$ as the excess of node i in P. When the context is clear, we use the shorter notations t_i and e_i.

Problem 3.5.5. Last excess problem Given a complete undirected graph $G = (V, E)$ with nodes $V = 1, \ldots, n$ and edges E, and parameters p_i and d_{ij} defined as in the orienteering problem, along with a positive bound M, the task is to find a path starting at node 1 that maximizes the sum of node profits while ensuring that the excess of the last node is at most M.

Lemma 3.5.6. *The last excess problem is approximable within a factor of* $2 + \varepsilon$.

Proof: To solve a last excess problem, we can consider an orienteering problem for each node $i \in V \setminus 1$ on the same graph with the same profits and edge times. The goal of each orienteering problem for node i is to find a 1-i path whose sum of edge times is bounded by $L = d_{1i} + M$. Suppose we have obtained a γ-approximation for each of the $n - 1$ orienteering problems. Then, the most profitable γ-approximation solution is a γ-approximation solution of the last excess problem. By relying on the $2 + \varepsilon$-approximation algorithm for the orienteering problem (Chekuri et al. 2012), we obtain the lemma. $\square$

For any path, $v_0(= 1) - v_1 - \cdots v_k$, if $i < j$, then $e_{v_j} = t_{v_j} - d_{1v_j} = t_{v_i} - d_{1v_i} + (d_{1v_i} + d_{v_i v_{i+1}} + \cdots + d_{v_{j-1} v_j} - d_{1v_j}) \geq t_{v_i} - d_{1v_i} = e_{v_i}$. This means that the later a node is visited on a path, the larger its excess becomes. Thus, if the excess of the last node of a path is bounded by a constant, the excess of every node on the path

is also bounded by the same constant. Moreover, we have $e_{v_j} - e_{v_i} = (d_{1v_i} + d_{v_i v_{i+1}} + \cdots + d_{v_{j-1}v_j} - d_{v_j})$, which is the excess of v_j of the subpath that visits v_i, v_{i+1}, $\ldots$, v_j excluding the starting node $v_0(= 1)$.

Lemma 3.5.7. *Let β be a integer with $\beta > \alpha$. Then, there exists a β-approximation solution P' of TRPP$_\alpha$ such that the excess at every node is at most $\alpha p/\beta$.*

Proof:

Let $P^* = v_0(= 1) - v_1 - \cdots v_k$ denote an optimal path of TRPP$_\alpha$. We consider a partition of the node set $\{v_1, v_2, \cdots, v_k\}$, and for each set of the partition, we consider the path that visits its nodes in the same order as in P^*, starting at node 1. We claim that there exists a partition with at most β sets, such that the excess of every node is no greater than $\alpha p/\beta$ in all the paths corresponding to its sets. Therefore, there must be a set in the partition whose path Q earns at least $1/\beta$ times the profit of P^*. Since each node of Q is visited no later than in P^*, Q is a β-approximation of TRPP$_\alpha$ that satisfies the condition in the lemma.

For each l, $1 \leq l \leq \beta$, let the set of nodes whose excesses belong to the interval $[(l-1)\alpha p/\beta, l\alpha p/\beta)$ be N_l. As the profit of every node is positive, the excess of each node cannot exceed αp. Therefore, if N_l is nonempty, their nodes should be consecutive in P^*. The sets N_l form a partition of the node set $\{v_1, v_2, \cdots, v_k\}$ from P^*. Let Q_l be the path that visits the nodes of N_l in the same order as in P^* except for node 1.

The difference between the excesses of the first and last nodes of Q_l is equal to the difference between the excesses of the first and last nodes of P^*, which is at most $\alpha p/\beta$. Since the excesses of nodes are monotone non-decreasing in the order of their visits, N_l's form the partition that we claimed. $\square$

Theorem 3.5.8. *For any $\varepsilon > 0$ and any integer $\beta > \alpha$, the approximation solution of the last excess problem in Lemma 3.5.6 is a $(2+\varepsilon)\beta^2/(\beta-\alpha)$-approximation of TRPP$_\alpha$ defined on the same G, r_i's and d_{ij}'s.*

Proof: Consider a path P from an approximation solution obtained from Lemma 3.5.6. If we can demonstrate that P is a $(2+\varepsilon)\beta/(\beta-\alpha)$-approximation solution of TRPP$_\alpha$ under the constraint that the excess of every node is no greater than $\alpha p/\beta$, then Lemma 3.5.7 implies the theorem.

In the restricted TRPP$_\alpha$, the profit from each node is between $p_i - \alpha p/\beta$ and p_i (inclusive). As p_i is in the range $[p, p\alpha)$ and $\alpha/\beta < 1$, p_i is at most $(1-\alpha/\beta)^{-1} = \beta/(\beta-\alpha)$ times the actual profit. Thus, the path P from a $(2+\varepsilon)$-approximation solution of the excess problem is a $(2+\varepsilon)\beta/(\beta-\alpha)$-approximation solution of TRPP$_\alpha$.

Theorem 3.5.8 asserts that TRPP$_\alpha$ admits a constant factor approximation when α is constant. If α is an integer, then the approximation factor is minimized when $\beta = 2\alpha$, and in that case, the approximation factor is $(8+\varepsilon)\alpha$.

We can use the aforementioned approximation algorithm for TRPP$_\alpha$ to construct an $O(\log n)$-approximation algorithm for TRPP. Given a TRPP instance, we first remove all nodes i with $p_i \leq \varepsilon \max_j p_j/n$. Since these nodes contribute at most $n\varepsilon \max_j p_j/n = \varepsilon \max_j p_j$ to the profit, deleting them cannot reduce the profit of any solution by more than a factor of ε times its objective value. Thus, we can guarantee that the objective value of any solution is at least $\max_j p_j$.

The ratio of the minimum to maximum p_i values among the remaining nodes is at most n/ε, so we can partition the set of remaining nodes into $O(\log n)$ subsets such that the restricted TRPP, in which a path can only visit nodes from a single subset of the partition, becomes a TRPP$_\alpha$ for some constant $\alpha > 0$. At least one

of the resulting TRPP_α instances has an optimal solution whose profit is at least $O(1/\log n)$ times the profit of an optimal solution of TRPP. Therefore, we obtain an $O(\log n)$-approximation solution to TRPP using the approximation algorithm for TRPP_α as a subroutine.

3.5.2 Hardness of Approximation

In this section, we prove that TRPP is max-SNP-hard, even when the node revenues are identical and d_{ij}'s are either 1 or 2. This implies that TRPP_2 is hard to approximate within an constant, that matches(up to constant) the approximation algorithm proposed in the previous section. let us denote the TSP, TRP, and TRPP, whose edge times are either 1 or 2, by TSP-$\{1,2\}$, TRP-$\{1,2\}$, and TRPP-$\{1,2\}$, respectively. We will reduce TSP-$\{1,2\}$ which is max-SNP-hard (Papadimitriou & Yannakakis 1993), to TRP-$\{1,2\}$, which we, in turn, reduce to TRPP-$\{1,2\}$.

We believe that the first reduction preserving APX-hardness from TSP-$1,2$ to TRP-$1,2$ was already discovered by Blum et al. (1994). Blum et al. (1994) remark that TSP-$1,2$ can be reduced to TRP, and a scheme similar to our reduction is described. let us make an observation. Suppose a node i of TRP-$\{1,2\}$ has $d_{ij} = 2$ with every adjacent node j. Suppose i is the k-th from last node of a path P. Let P' be a path obtained by moving node i to the last node in P. Comparing the sum of the arrival times at the nodes of P and P', we can observe that the arrival time at each of the $k-1$ nodes after i of P decreases by at least 2, and the the arrival time at node i increases by at most by $2(k-1)$ in P'. Therefore, the sum of the arrival times at the nodes of P' is no larger than the one from P. The reduction from TSP-$\{1,2\}$ to TRP-$\{1,2\}$ relies on the observation.

Let $G = (V, E)$ be a graph that represents a TSP-$\{1, 2\}$ problem. It is known that this problem cannot be approximated within a factor of $1+\alpha$, where $\alpha = 1/740$ (Engebretsen & Karpinski 2001). Let n be the number of nodes in G. We define G' as the graph obtained by adding nM nodes to G, where each added node is connected to every other node with an edge of distance 2. We denote the set of added nodes as W, and set $M = 1481$, which is a value similar to $2/\alpha$. If the optimal distance of TSP-$\{1, 2\}$ is T, then there exists a Hamiltonian path of G starting at node 1 with a distance of at most $T - 1$. We can construct a path P in G' that follows this Hamiltonian path and then visits the nodes of W in an arbitrary order.

Suppose TRP-$\{1, 2\}$ can be approximated within a factor $1 + \beta$, where $\beta = 1/2191881$, a value similar to $4/\alpha^2$. Let Q be a path obtained by applying the $1+\beta$-approximation algorithm to G'. We then reorder the nodes of Q while maintaining the order among the nodes of V so that the nodes of W are visited later than the nodes of V. From the above observation, this does not increase the sum of the arrival times at the nodes. Let the new path be Q' and the length of the subpath consisting of the nodes of V be L. Then the objective value of Q' is at least $(L+2+(nM+1))nM$.

Since Q' is a $1 + \beta$-approximation of TRP-$\{1, 2\}$ on G', its objective value is no greater than the total arrival times at each node of P as a solution of the same TRP. It is not difficult to see that the sum of the arrival times at the nodes of P is no greater than $((T-1)+2+(Mn+1))Mn+nT$. Thus we have $(L+2+(Mn+1))Mn \leq (1 + \beta)\left(((T - 1) + 2 + (Mn + 1))Mn + nT\right)$.

Then we get:

$$\begin{aligned} L + 3 &\leq T(1+\beta)(1+1/M) + \beta M n + 2(1+\beta) \\ &\leq T(1+\beta+1/M+\beta/M+\beta M) + 2(1+\beta), \end{aligned} \tag{3.9}$$

where the second inequality in (3.9) follows from $T \geq n-1$. If $\alpha = 1/740$, $\beta = 1/2191881$, and $M = 1481$, then $(\beta+1/M+\beta/M+\beta M) < \alpha$. Since $L, T \geq n-1$, for sufficiently large n, we have $L+2 \leq (1+\alpha)T$. The left-hand side of this inequality provides an upper bound on the subpath consisting of the nodes of V plus the distance of the last edge returning to node 1. Therefore, the inequality contradicts the fact that a $1+\alpha$-approximation is impossible for TSP-$\{1,2\}$.

Theorem 3.5.9. *TRP-$\{1,2\}$ is NP-hard to approximate within a factor of $\frac{2191882}{2191881}$.*

We now reduce the TRP-$\{1,2\}$ to TRPP-$\{1,2\}$. To do this, let $G = (V, E)$ be the graph of TRP-$\{1,2\}$. We consider the TRPP-$\{1,2\}$ on the same graph $G = (V, E)$ with every node assigned the identical revenue $2n-2$. Suppose a path P does not visit a node $i \neq 1$. Then, by visiting i after P, it will earn a profit at least $2n-2-2(n-2) = 2$. This means that an optimal path is Hamiltonian and its objective value is $2n(n-1)$ minus its objective value as a solution of TRP-$\{1,2\}$.

Suppose TRPP-$\{1,2\}$ is $(1+\alpha)$-approximable. The objective value of TRP-$\{1,2\}$ is no less than $n(n-1)/2$ and no greater than $n(n-1)$. Therefore, the corresponding objective value of TRPP-$\{1,2\}$ is no less than $n(n-1)$ and no greater than $3n(n-1)/2$. A $(1+\alpha)$-approximation solution of TRPP-$\{1,2\}$ is no less than the optimum by more than $3\alpha n(n-1)/2$, which is in turn no greater than 3α times the optimum of TRP-$\{1,2\}$. Therefore, a $1+\alpha$-approximation of TRPP-$\{1,2\}$ is a $1+3\alpha$-

approximation of TRP-$\{1, 2\}$. Combining this result with the inapproximability of TRPP-$\{1, 2\}$, we can derive the following theorem.

Theorem 3.5.10. *Even with uniform node revenues, TRPP-$\{1, 2\}$ is NP-hard to approximate within a factor of $\frac{6575644}{6575643}$.*

3.6 Conclusion

We have shown the sMTP is approximable within the factor $O\Big(n^{1/2} \log n \min\Big\{ \log p_{\max}, \max\big\{ \log m, \log^{\delta} \min\{p_{\max}, d_{\text{ratio}}\}\big\}\Big\}\Big)$ whereas inapproximable within $n^{\log^{-\varepsilon} n}$ times the optimum. We also show that the integrality gap is $\theta(n^2)$. As byproducts, We gives approximation algorithms for ssMTP and intermediate problems, t-intersecting and t-separable sMTP. We also provide approximation algorithms for TRPP and TRPP$_\alpha$, with approximation ratio $O(\log n)$ and $O(1)$, respectively.

Chapter 4

Comparative Experiments

4.1 Introduction

In Chapter 2, we proposed two formulations of sMTP, the Arc-Flow Formulation and the Triple Formulation, and additional inequalities with corresponding separation algorithms. In Chapter 3, we proposed an approximation algorithm for sMTP. In this chapter, we conduct experiments to compare the performance of the previously proposed modes and algorithms.

First, we examine which inequalities should be initially added to the formulation, considering the wide range of options available for adding inequalities to the TF.

Next, we compare the performance of the branch-and-bound method for AF and the improved TF, along with the cut-and-branch method, which involves separating 3-Criteria inequalities and 3-Criteria-TF inequalities at the root nodes of AF and the improved TF, respectively.

Finally, we compare heuristic algorithms. We compare an algorithm that has been modified from the approximation algorithm in Chapter 3 to be more practical, and two basic heuristic algorithms for contrasting purposes.

4.2 Experimental Setting

The hardware and operating system used for this study were an Intel(R) Core(TM) i5-6400 CPU @ 2.70 GHz PC with 8GB RAM and Windows 10 Home (64 bits). The programming language and library for mathematical formulation and LP solver used were Python and CPLEX 22.1.1.

We used randomly generated instances, where each of the n nodes was assigned random coordinates in a two-dimensional Euclidean space. The first component of node i is $100i/n$, while the other components were uniformly random integers in the interval $[0, 100]$.

In the Backhaul Profit Maximization Problem, the positions of nodes 1 and n are predetermined, while the positions of other nodes are randomly chosen within an ellipse where the sum of distances from nodes 1 and n, acting as focuses, does not exceed the upper bound of the total distance (Dong et al. 2022).

Considering the fixed order of nodes, one might argue that it is suitable to assume a situation where the locations of each node spread out in one direction when thinking about a railway network. In this case, we can consider a situation where the ratio between $c_{ij} + c_{jk}$ and c_{ik} is close to 1 for any three nodes i, j, k. However, it is unnecessary to specifically consider such cases because it is possible to amplify the ratio between $c_{ij} + c_{jk}$ and c_{ik}. By adding or subtracting a constant value to the c_{ij} and $r(i, j)$ values where $i \leq t < j$ for a certain t, the objective function for the same set of feasible solutions remains unchanged since it is the difference between revenue and cost. Therefore, even if, for example, $c_{12} = 51$, $c_{23} = 50$, and $c_{13} = 100$, the ratio $\frac{c_{12}+c_{23}}{c_{13}} = 1.01$ is close to 1, we can still adjust the values through the aforementioned modification to make $c_{12} = 1$, $c_{23} = 0$, and $c_{13} = 0$. In this case, the

ratio becomes infinitely large.

Initially, experiments were conducted with instances where the ratio of length to width was different from $[0, 100] \times [0, 100]$. However, it was observed that this did not change the difficulty of the problem. However, it was noted that the performance of each methodology varied significantly depending on the revenue setting. Therefore, various scenarios were considered. In general, the revenue was set to increase as the cost increased using three different approaches. Firstly, following the case of the Backhaul Profit Maximization Problem, the revenue was simply set to 1.2 times the cost. In this case, the parameter rctype is denoted as A.

Secondly, each revenue $r(k, l)$ was set as the product of a randomly generated value X_{kl}, drawn from a uniform distribution $uniform[0, \alpha]$, and the cost c_{kl} between the corresponding nodes. Here, α was chosen from the set $2, 3$. If X_{kl} became less than 1, it was considered as if there was no corresponding product. In this case, the parameter rctype is denoted as B.

Thirdly, the revenue was set as the sum of a value multiplied by 1 and $[0, (k-l)\beta]$. Here, β was chosen from the set $2, 3$. This configuration encourages long-distance transportation and results in higher profits for transportation between cities with larger differences in indices. In this case, the parameter rctype is denoted as C.

Finally, the capacity U was set to a value obtained by multiplying the capacity that could accommodate all trades by the parameter uratio $\in \{0.05, 0.2, 1\}$. Five instances were generated for each quadruple $(n, \text{rctype}, \text{uratio})$.

4.3 Computational Comparison of Relaxation Bounds

The following are the candidate inequalities to be initially added to (TF).

$$x_{kl} \leq d(k,l)(\sum_{k<i\leq l} y_{kj}) \; \forall k < l \tag{4.1a}$$

$$x_{kl} \leq d(k,l)(\sum_{k\leq i<l} y_{il}) \; \forall k < l \tag{4.1b}$$

$$x_{kl} \leq d(k,l)(\sum_{k\leq i\leq t<j\leq l} y_{ij}) \; \forall k \leq t < l \tag{4.1c}$$

$$u_{ij}^{l} \leq d([1,i],l)y_{ij}, \; \forall i < j \leq l \tag{4.1d}$$

(4.1c) includes both (4.1a) and (4.1b). (TF1) to (TF6) represent the addition of (4.1b), (4.1a), (4.1a,4.1b), (4.1a, 4.1b, 4.1d), (4.1c), and (4.1c, 4.1d) to (TF), respectively.

We will use the LP gap (%): $(\text{OPT}_{LP} - \text{OPT})/\text{OPT}$ to measure the tightness of the bound between the optimal objective value OPT of the original problem and the optimal objective value OPT_{LP} of each LP relaxation.

We have shown that the (AF) is tighter than (TF). The experimental results are shown below. The average LP gap is computed for each case. Instances generated with $n = 20$ and $n = 25$, the average LP gap for each formulation was as follows:

The average computation times were as follows:

We first examined whether it is beneficial to add (4.1d) to the formulation. Comparing (TF3) and (TF4), the average computation time of (TF4) was found to be 43% faster and it provided faster answers in 69% of cases. Similarly, comparing (TF5) and (TF6), the average computation time of (TF6) was 43% faster, and (4.1d) yielded faster answers in 92% of cases. Based on these two analyses, it can be concluded that adding (4.1d) to the formulation is generally beneficial. It is

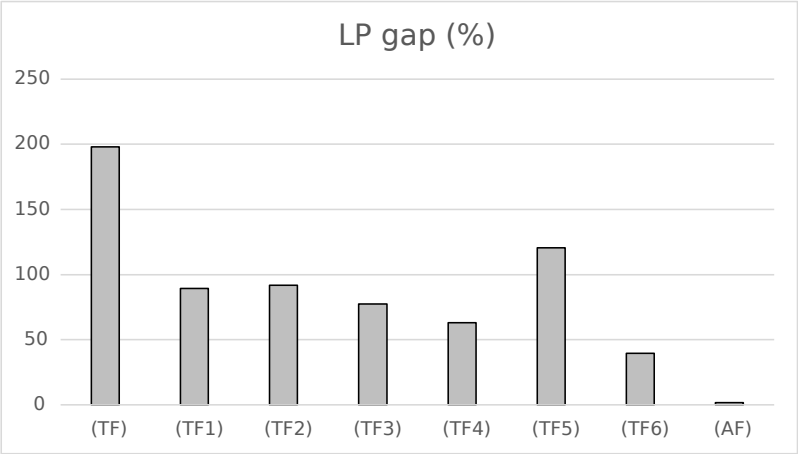

Figure 4.1: LP gap comparison for different formulations

interesting to note that the decrease in average computation time was similar at 43%.

Next, an analysis was conducted to determine which among (4.1a)-(4.1c) should be added to the formulation. It was found that including (4.1c) resulted in poorer performance of (TF5) and (TF6) compared to (TF3) and (TF4), mainly due to increased computation time for each subproblem. As for whether to include (4.1a) or (4.1b), it was observed that including both in (TF3) resulted in the fastest average computation time and outperformed the others in 81% of cases. Consequently, we decided to adopt (TF4) as the base formulation.

While the LP gap and computation time among instances based on (TF) generally maintained a consistent relationship with the overall average, the performance comparison between the (TF) formulation and the (AF) formulation varied significantly depending on the instance. On average, the (AF) formulation outperformed (TF), but this trend varied greatly depending on the instance. Comparing it with

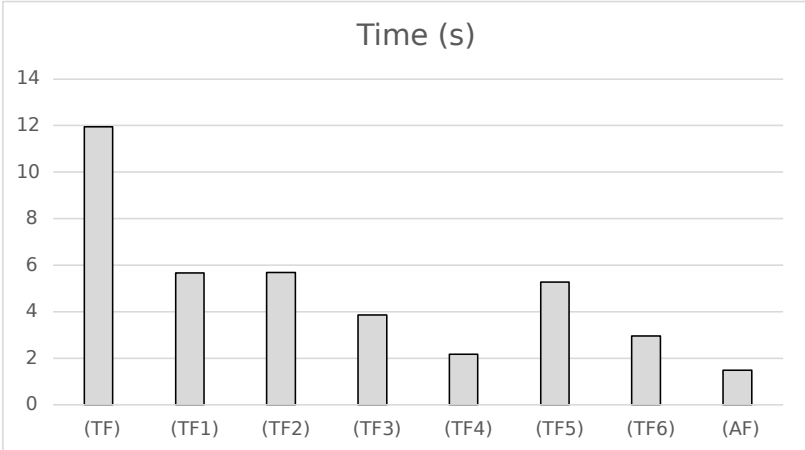

Figure 4.2: Computation time comparison for different formulations

the best-performing variant of (TF), (TF4), we observe the following:

For (TF4), the average computation time was $(2.56, 2.22, 1.09)$ when the rctype parameter was set to (A, B, C) respectively. In contrast, (AF) consumed computation times of $(0.14, 0.18, 4.3)$ for the same parameter values. It is evident that (AF) particularly incurred longer computation times for instances with an rctype of C. Upon examining the data generation method, it is apparent that instances with an rctype of C have a higher ratio of revenue to cost compared to the other two cases. Note that compared to (AF), (AF) with 3-Criteria inequalities showed an average decrease of LP gap by about 52%.

4.4 Comparison of Computation time Based on the use of Cut

In this section, we compare the computation times when applying the branch-and-bound method to (AF) and (TF) with the addition of the separation process for the

3-Criteria inequality and 3-Criteria-TF inequality at the root node. We conducted the comparison for cases where the number of nodes ranges from 30 to 35. We only considered cases where rctype is either B or C.

The distinctive feature of the 3-Criteria inequality and 3-Criteria-TF inequality is that both the left-hand side and right-hand side of the equations are less than or equal to 1. In other words, the degree to which these inequalities violate is not more than 1. We set the tolerance value to 0.1, which means that we added the cut obtained from the separation process if the incumbent solution violates the inequality by more than 0.1.

Let's denote TF4 as the method of conducting branch-and-bound on (TF4), and TF4C as TF4 with the additional cut-and-branch process. For the cases where $n = 30$ and $n = 35$, with a time limit of 600s, the following results were obtained:

Table 4.1: Test result with rctype $= C$

Parameters			TF4		TF4C		
n	rctype	uratio	num of nodes	time	num of nodes	time	num of cuts
30	C	0.05	44.7	11.4	5.5	11.3	6
		0.2	31.7	16	4.5	16	9.2
		1	572	63.1	108.2	70	144
35	C	0.05	16	16	3	12	4
		0.2	42	41	5	59	34
		1	1556	320	247	261	211

For $n = 30$, the number of branch trees decreased on average by a geometric mean of 4.4 times, while the average computation time increased from 30s to 32s, resulting in a 7.5% increase. For $n = 35$, the number of branch trees decreased on average by a geometric mean of 1.9 times, while the average computation time decreased from 107s to 96s, resulting in a 11% decrease. For the algorithm with (AF),

the computation time exceeded an average of 300 seconds when $n = 30$. In the case of $n = 35$, algorithms with (AF) exceeded the time limit of 600s when solving the LP at the root node for instances with rctype $= B$ and uratio $= 0.05, 0.2$. However, for instances with uratio $= 1$, the LP was solved successfully, but the optimal solution could not be obtained within the given time limit.

Table 4.2: Test result with $n = 35, \text{rctype} = B$

TF4		TF4C			AF	
nodes	time	nodes	time	cuts	nodes	time
16626	443	10099	374	39	21	101

In cases where rctype is B, AF demonstrated superior performance compared to TF4C. Additionally, TF4C solved problems faster in 94% of cases compared to TF4. Considering cases where rctype is C, we can expect TF4C to outperform TF4 in solving problems with $n \geq 35$.

An iteration of adding a cut to (TF) takes 0.5 seconds on average, with separation accounting for 0.4 seconds of that time. This corresponds to twice the duration when $n = 30$, and since $(35/30)^4 \simeq 1.85$, we can once again confirm that the computation time is proportional to $O(n^4)$. The separation of 3-Criteria inequality in AFC takes approximately 1.1 seconds for $n = 30$ and 2.2 seconds for $n = 35$ on average. When a every cut is added with a threshold of 0.3, it reduces the number of explored nodes by around 20%. However, the total computation time increases in most cases. This can be attributed to the relatively small size of the branching tree in AF, consisting of several hundred nodes, and the LP gap is quite tight as observed.

4.5 Comparison of Heuristic Algorithms

In this section, we present three heuristic algorithms and compare the performance of three heuristic algorithms.

Heuristic from Approximation Algorithm

The first approach is an algorithm that utilizes the idea of the approximation algorithm that proposed in Chapter 3.

1. Let $f(i,j) = 0$, For each $i < j$.
2. Repeat the following for each $i < j$ and $k \in 0, \cdots, j-i-1$:
   - 2-1. Find the i-j shortest path $p(i,j,k)$ that passes through k nodes between i and j.
   - 2-2. Determine the maximum profit achievable through trades within each path $p(i,j,k)$. If this profit exceeds $f(i,j)$, update the value of $f(i,j)$ with this profit.
3. Output the length of the longest path from 1 to n, where the cost of each arc is $f(i,j)$.

In the approximation algorithm in Chapter 3, we divided the original problem into t-intersecting subproblems defined by vertices i to j and patched them up accordingly. Each t-intersecting problem was further divided by selecting the shortest path, among the paths passing through a specific number or fewer vertices according to Algorithm 3.3.12 or 3.3.8, prior to t. These divided subproblems became ssMTPs, with an FPTAS computation time of $O(n^7)$ for the ssMTP and a pseudo-polynomial time algorithm that includes a term $c_{\max}$, representing the maximum cost among

the terms U and the costs associated with the arcs, making it difficult to solve within a reasonable computation time. Therefore, we aimed to fix both sides, rather than just one side, of t by selecting sufficiently good paths. The shortest path between two nodes among those passing through a specific number or fewer vertices includes the one with the minimum distance between any two vertices passing through the same number of vertices. From this reasons, in step 2-1, we calculate $p(i, j, k)$ and compute the maximum profit obtainable within that path.

Randomized Rounding

The second approach is randomized rounding. After solving the LP relaxation of (AF), we select each arc ij with the probability corresponding to the value of y_{ij} in the solution. We obtain a feasible 1-n path and then find the optimal x value along that path. In the experiments, we increased the number of iteration N from 100 to 500 as n increased.

Algorithm 4.5.1.

1. *Solve the linear relaxation problem of the (AF) to obtain the optimal solution* $(\bar{x}, \bar{y})$.

2. *Repeat the following steps* N *times to obtain a set of* 1-n *paths* $p_1, \ldots, p_N$.

   2-1. Set $V(p) \leftarrow \{1\}$ *and* $i \leftarrow 1$.

   2-2. While $i \neq n$*, do the following:*

   - *Select* j' *with probability* $\frac{\bar{y}_{ij}}{\sum_{m=i+1}^{n} \bar{y}_{im}}$ *for* $j \in i+1, \ldots, n$.
   - *Update* $V(p)$ *by adding* j'*, and set* $i \leftarrow j'$.

3. *For each $p \in \{p_1, \ldots, p_N\}$, fix the path represented by y in the arc formulation to p and solve the LP to obtain the optimal $x = x_p^*$.*

4. *Output the solution with the largest objective function among x_p^*, where $p \in \{p_1, \ldots, p_N\}$.*

2-node Path

The third candidate approach is a naive method where we fix each path to visit a maximum of two nodes, excluding nodes 1 and n. It solves the optimization problem (AF) after the path is fixed, and outputs the best feasible solution obtained. This approach was performed solely for the purpose of comparing computation times. The quality of the output improves as the number of visited nodes decreases in the optimal solution. However, in the worst case, it may output a feasible solution that differs by a factor of $O(n^2)$. As shown in Chapter 3, this is equivalent to the ratio of the objective function value between the worst-case optimal solution and the randomized rounding algorithm.

For $n = 20$, the three methods require $(9s, 5s, 7s)$ as a baseline, and in the case of randomized rounding, the additional time is needed to solve the relaxation of (AF). For $n = 25$, $n = 30$, and $n = 40$, excluding the time to solve (AF) in randomized rounding, the methods require $(24s, 9s, 18s)$ and $(54s, 29s, 41s)$, $(212s, 54s, 152s)$ respectively.

Table 4.3 summarizes the average gap between the objective function and the optimal solution for each method based on the rctype when $n = 20$ and $n = 30$. The gap with the objective function is calculated as $\frac{b-a}{b}$, where a is the objective

Table 4.3: Comparison on the gap between method outputs and optimal solutions

n	rctype	ApproHeur	RanRound
20	A	0.6%	0.008%
	B	4%	0.05%
	C	4%	0.6%
30	A	1.9%	0%
	B	11%	12%
	C	10%	3%
40	A	1.6%	3%
	B	27%	37%
	C	24%	8%

function value obtained and b is the optimal objective function value. ApproHeur refers to the first method, while RanRound represents the second method, and each cell denotes the average gap. We cannot consider the 2-node path as producing good solutions, as even for $n = 20$, the average gap for each type exceeds 35%.

Note that the solutions from the 2-node Path has, on average, a 41% gap compared to the optimum, even when considering $n = 20$. As n increases, ApproHeur is likely to yield better results in terms of both time and solution quality compared to RanRound. Firstly, when n is 30, the average time required to solve the LP increased by 51s compared to the case when n was 20, resulting in a total of 80s. This indicates that as n grows larger, the computation time for ApproHeur increases even more compared to RanRound, potentially exceeding practical time limits. Additionally, ApproHeur exhibits smaller variations in the gap depending on the value of rctype, while RanRound demonstrates significant gap fluctuations when rctype is C. For $n = 30$, the ApproHeur algorithm outputted better solutions for 53% of the instances with rctype $= C$. Therefore, for instances with rctype being C, ApproHeur is more likely to produce better solutions for larger values of n.

Chapter 5

Conclusion

5.1 Summary of contributions

This study dealt with the Single Path Multicommodity Trading Problem on Acyclic Network (sMTP). Two mathematical models were proposed for this problem, and valid inequalities were found for each of them. Some of these were shown to have necessary and sufficient conditions for facet-defining. The upper bounds obtained from the linear programming relaxation problems for each model and their models with added valid inequalities were compared theoretically and experimentally. In addition, the approximability of this problem was addressed. In this context, while inapproximability of sMTP and the integrality gap of the proposed mathematical models were presented, approximation algorithms were also proposed. In the process, approximation algorithms for special cases of this problem and related problems were also presented. Since research on facet-defining inequalities and approximation algorithms related to this problem was lacking, the results of this study have the potential to be extended and applied to related problems.

In Chapter 2, two formulations of sMTP, the Arc-Flow Formulation and the Triple Formulation, were presented. The dimensions of each were determined, and conditions that facet-defining inequalities must satisfy were obtained through pro-

jection from perfect extended formulations. A valid inequality called the 3-Criteria Inequality was presented for the Arc-Flow Formulation, and necessary and sufficient conditions for this to be a facet-defining inequality and a separation algorithm were provided. In addition, other classes of inequality were generalized from the 3-Criteria Inequality, and classes of valid inequalities for the Triple Formulation were also presented.

In Chapter 3, it was shown that sMTP is a generalization of the Max-Rep problem, and therefore, sMTP has the same inapproximability as Max-Rep. Furthermore, it was shown that the integrality gap of the formulations introduced in Chapter 2 is $\theta(n^2)$, indicating that it cannot provide a better approximation algorithm more than considering just select only one request. To obtain an approximation algorithm for sMTP, approximation algorithms were obtained for ssMTP and t-separable sMTP. The approximation ratio of the basic t-separable sMTP approximation algorithm can be as large as $\Omega(n^2)$ depending on the scale of the parameter, but a boosting method was proposed to improve it and reduce the parameter-related factor. Based on the relationship that the approximation ratio of sMTP is $O(\log n)$ times that of t-separable sMTP, an approximation algorithm for sMTP was obtained. Finally, approximation algorithms were obtained for problems similar to sMTP, including the Traveling Repairman Problem with Profits (TRPP), and the constant upper and lower bounds for the approximation ratio of the special case of TRPP, TRPP_α, were obtained.

In Chapter 4, we present basic approaches for solving the sMTP using the results from Chapters 2 and 3, and verify their practical solvability on various instances. First, we consider various methods for generating instances to conduct experiments.

We compare the quality of upper bounds obtained by solving the LP relaxation problem when we add the discovered classes of valid inequalities to the Arc-Flow Formulation (AF) and Triple Formulation (TF) proposed in Chapter 2. From this comparison, we determine the set of inequalities that should be added to the formulation from the beginning. Next, we compare the performance of a branch-and-bound algorithm using the AF and TF models with valid inequalities added, considering the 3-Criteria Inequalities and 3-Criteria-TF Inequalities found in Chapter 2 as cuts at each root node, using a cut-and-branch algorithm. Lastly, we propose a heuristic algorithm that adopts the idea of the approximation algorithm proposed in Chapter 3. We conduct comparative experiments between this heuristic algorithm and an algorithm that applies randomized rounding to the LP relaxation solution of AF.

5.2 Further Research Directions

We have confirmed that there can be various valid inequalities when the f-variable of sMTP has negative coefficients. Therefore, in this case, it is possible to consider lifting existing valid inequalities according to the problem situation. Furthermore, when considering more valid inequalities for sMTP, it is necessary to find heuristic separation algorithms, as there may not be an exact polynomial-time separation algorithm, or the computation time may be too high, such as $O(n^8)$.

On the other hand, other formulations can also be considered. Although not shown in this study, the path formulation using paths as variables and the node formulation using binary variables to indicate whether a node is traversed or not did not yield good results.

There are two directions to obtain a better approximation algorithm for sMTP.

One is to make efforts to remove terms related to parameters such as demand or revenue from the approximation ratio. The other is to reduce the exponential term of n, including the possibility of applying Chalermsook et al. (2012)'s $O(n^{1/3})$-approximation algorithm for Max-Rep. However, at present, this algorithm is based on a solution that seems to have no corresponding concept in sMTP, such as finding the maximum matching in Max-Rep.

In addition, there may be a direction to perform two studies on sMTP for more generalized problems. In reality, there are various situations where additional constraints need to be considered. One important direction is to extend sMTP to cases where the underlying graph can be cyclic. If we simply extend it, we can easily see that satisfying each request one by one in the shortest distance is optimal, as there is no cost incurred when moving to the empty state. Therefore, we need to consider various additional conditions, such as upper bounds on the total distance traveled or time windows for each request. When a cyclic graph exists, the relationship with t-separable instances becomes unclear. It is possible to decompose the optimal solution into $O(\log n)$ solutions, but we do not know the instances of each decomposed problem.

Bibliography

Agarwal, Y. K. (2018), 'Network loading problem: Valid inequalities from 5-and higher partitions', *Computers & Operations Research* **99**, 123–134.

An, H.-C., Kleinberg, R. & Shmoys, D. B. (2015), 'Improving christofides' algorithm for the st path tsp', *Journal of the ACM (JACM)* **62**(5), 1–28.

Andrews, M., Chuzhoy, J., Khanna, S. & Zhang, L. (2005), Hardness of the undirected edge-disjoint paths problem with congestion, *in* '46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)', IEEE, pp. 226–241.
URL: *https://doi.org/10.1109/sfcs.2005.41*

Arora, S., Babai, L., Stern, J. & Sweedyk, Z. (1997), 'The hardness of approximate optima in lattices, codes, and systems of linear equations', *Journal of Computer and System Sciences* **54**(2), 317–331.
URL: *https://doi.org/10.1006/jcss.1997.1472*

Arora, S. & Lund, C. (1996), *Hardness of Approximations*, PWS Publishing Co., USA, pp. 399–446.

Atamtürk, A., Gómez, A. & Küčükyavuz, S. (2016), 'Three-partition flow cover inequalities for constant capacity fixed-charge network flow problems', *Networks* **67**(4), 299–315.
URL: *https://doi.org/10.1002/net.21677*

Atamtürk, A., Küčükyavuz, S. & Tezel, B. (2017), 'Path cover and path pack inequalities for the capacitated fixed-charge network flow problem', *SIAM Journal on Optimization* **27**(3), 1943–1976.
URL: *https://doi.org/10.1137/15m1033009*

Balas, E. (1989), 'The prize collecting traveling salesman problem', *Networks* **19**(6), 621–636.

Balas, E. (1998), 'Projection with a minimal system of inequalities', *Computational Optimization and Applications* **10**, 189–193.

Balcan, M.-F. & Blum, A. (2006), Approximation algorithms and online mechanisms for item pricing, *in* 'Proceedings of the 7th ACM Conference on Electronic Commerce', pp. 29–35.
URL: *https://doi.org/10.1145/1134707.1134711*

Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P. & Sudan, M. (1994), The minimum latency problem, *in* 'Proceedings of the 26th Annual ACM SIGACT Symposim on Theory of Computing', pp. 163–171.

Blum, A., Chawla, S., Karger, D. R., Lane, T., Meyerson, A. & Minkoff, M. (2007), 'Approximation algorithms for orienteering and discounted-reward tsp', *SIAM Journal on Computing* **37**(2), 653–670.
URL: *https://doi.org/10.1137/050645464*

Borndörfer, R., Grötschel, M. & Pfetsch, M. E. (2007), 'A column-generation approach to line planning in public transport', *Transportation Science* **41**(1), 123–

132.

URL: *https://doi.org/10.1287/trsc.1060.0161*

Borndörfer, R., Grötschel, M. & Pfetsch, M. E. (2008), *Models for line planning in public transport*, Springer.

Chakrabarti, A., Chekuri, C., Gupta, A. & Kumar, A. (2007), 'Approximation algorithms for the unsplittable flow problem', *Algorithmica* **47**(1), 53–78.

Chalermsook, P., Chuzhoy, J., Kannan, S. & Khanna, S. (2012), Improved hardness results for profit maximization pricing problems with unlimited supply, *in* 'Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques', Springer, pp. 73–84.

URL: *https://doi.org/10.1007/978-3-642-32512-0_7*

Charikar, M., Hajiaghayi, M. & Karloff, H. (2011), 'Improved approximation algorithms for label cover problems', *Algorithmica* **61**(1), 190–206.

URL: *https://doi.org/10.1007/s00453-010-9464-3*

Chekuri, C. & Khanna, S. (2007), 'Edge-disjoint paths revisited', *ACM Transactions on Algorithms (TALG)* **3**(4), 46.

URL: *https://doi.org/10.1145/1290672.1290683*

Chekuri, C., Khanna, S. & Shepherd, F. B. (2006), 'An $o(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow', *Theory of Computing* **2**(7), 137–146.

URL: *https://theoryofcomputing.org/articles/v002a007*

Chekuri, C., Khanna, S. & Shepherd, F. B. (2013), 'The all-or-nothing multicommodity flow problem', *SIAM Journal on Computing* **42**(4), 1467–1493.
URL: *https://doi.org/10.1137/100796820*

Chekuri, C., Korula, N. & Pál, M. (2012), 'Improved algorithms for orienteering and related problems', *ACM Transactions on Algorithms (TALG)* **8**(3), 1–27.
URL: *https://doi.org/10.1145/2229163.2229167*

Cheng, G. & Ansari, N. (2004), 'Finding all hops shortest paths', *IEEE Communications Letters* **8**(2), 122–124.
URL: *https://doi.org/10.1109/lcomm.2004.823365*

Chlamtáč, E., Manurangsi, P., Moshkovitz, D. & Vijayaraghavan, A. (2017), Approximation algorithms for label cover and the log-density threshold, *in* 'Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms', SIAM, pp. 900–919.
URL: *https://doi.org/10.1137/1.9781611974782.5*

Chouman, M., Crainic, T. G. & Gendron, B. (2017), 'Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design', *Transportation Science* **51**(2), 650–667.
URL: *https://doi.org/10.1287/trsc.2015.0665*

Chuzhoy, J. & Kim, D. H. K. (2015), On approximating node-disjoint paths in grids, *in* N. Garg, K. Jansen, A. Rao & J. D. P. Rolim, eds, 'Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)', Vol. 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH,

Wadern/Saarbruecken, Germany, Dagstuhl, Germany, pp. 187–211.
URL: *http://drops.dagstuhl.de/opus/volltexte/2015/5303*

Chuzhoy, J. & Li, S. (2016), 'A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2', *Journal of the ACM (JACM)* **63**(5), 1–51.
URL: *https://doi.org/10.1145/2893472*

Coene, S. & Spieksma, F. C. (2008), 'Profit-based latency problems on the line', *Operations Research Letters* **36**(3), 333–337.
URL: *https://doi.org/10.1016/j.orl.2007.11.001*

Cordeau, J.-F. (2006), 'A branch-and-cut algorithm for the dial-a-ride problem', *Operations Research* **54**(3), 573–586.

Dantzig, G. & Fulkerson, D. R. (2003), 'On the max flow min cut theorem of networks', *Linear inequalities and related systems* **38**, 225–231.

Dewilde, T., Cattrysse, D., Coene, S., Spieksma, F. C. & Vansteenwegen, P. (2013), 'Heuristics for the traveling repairman problem with profits', *Computers & Operations Research* **40**(7), 1700–1707.
URL: *https://doi.org/10.1016/j.cor.2013.01.003*

Dong, Y., Bai, Y., Olinick, E. & Yu, A. J. (2022), 'The backhaul profit maximization problem: Optimization models and solution procedures', *INFORMS Journal on Optimization* **4**(4), 373–402.
URL: *https://doi.org/10.1287/ijoo.2022.0071*

Dong, Y., Olinick, E. V., Jason Kratz, T. & Matula, D. W. (2014), 'A compact linear programming formulation of the maximum concurrent flow problem', *Networks*

65(1), 68–87.

URL: *https://doi.org/10.1002/net.21583*

Engebretsen, L. & Karpinski, M. (2001), Approximation hardness of tsp with bounded metrics, *in* 'Automata, Languages and Programming: 28th International Colloquium, ICALP 2001 Crete, Greece, July 8–12, 2001 Proceedings 28', Springer, pp. 201–212.

Feillet, D., Dejax, P. & Gendreau, M. (2005), 'Traveling salesman problems with profits', *Transportation science* **39**(2), 188–205.

URL: *https://doi.org/10.1287/trsc.1030.0079*

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X. & Koenig, S. (2013), 'Ridesharing: The state-of-the-art and future directions', *Transportation Research Part B: Methodological* **57**, 28–46.

URL: *https://doi.org/10.1016/j.trb.2013.08.012*

Goemans, M. X. (1995), 'Worst-case comparison of valid inequalities for the tsp', *Mathematical Programming* **69**(1-3), 335–349.

Grötschel, M., Lovász, L. & Schrijver, A. (2012), *Geometric algorithms and combinatorial optimization*, Vol. 2, Springer Science & Business Media.

Gu, Z., Nemhauser, G. L. & Savelsbergh, M. W. (1999), 'Lifted flow cover inequalities for mixed 0-1 integer programs', *Mathematical Programming* **85**(3), 439–467.

URL: *https://doi.org/10.1007/s101070050067*

Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B. & Yannakakis, M. (2003), 'Near-optimal hardness results and approximation algorithms for edge-

disjoint paths and related problems', *Journal of Computer and System Sciences* **67**(3), 473–496.
URL: *https://doi.org/10.1016/s0022-0000(03)00066-7*

Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M., Petering, M. & Tou, T. W. (2018), 'A survey of dial-a-ride problems: Literature review and recent developments', *Transportation Research Part B: Methodological* **111**, 395–421.
URL: *https://doi.org/10.1016/j.trb.2018.02.001*

Jafari, E., Rambha, T., Khani, A. & Boyles, S. D. (2016), The for-profit dial-a-ride problem on dynamic networks, *in* 'the 95th Annual Meeting of the Transportation Research Board, Washington, DC', number 16-4686.

Kawarabayashi, K.-i., Kobayashi, Y. & Reed, B. (2012), 'The disjoint paths problem in quadratic time', *Journal of Combinatorial Theory, Series B* **102**(2), 424–435.
URL: *https://doi.org/10.1016/j.jctb.2011.07.004*

Kim, J.-E. (2015), 'Np-hardness of bubosang problem', *unpulished manuscript* .

Kleinberg, J. M. (1996), Approximation algorithms for disjoint paths problems, PhD thesis, Massachusetts Institute of Technology.

Kortsarz, G. (2001), 'On the hardness of approximating spanners', *Algorithmica* **30**(3), 432–450.
URL: *https://doi.org/10.1007/s00453-001-0021-y*

Lee, K., Park, K. & Park, S. (2009), 'A compact formulation of the ring loading problem with integer demand splitting', *Operations research letters* **37**(2), 111–113.

Lekkeikerker, C. & Boland, J. (1962), 'Representation of a finite graph by a set of intervals on the real line', *Fundamenta Mathematicae* **51**, 45–64.

Lu, Y., Benlic, U., Wu, Q. & Peng, B. (2019), 'Memetic algorithm for the multiple traveling repairman problem with profits', *Engineering Applications of Artificial Intelligence* **80**, 35–47.
URL: *https://doi.org/10.1016/j.engappai.2019.01.014*

Ma, B. & Wang, L. (2000), 'On the inapproximability of disjoint paths and minimum steiner forest with bandwidth constraints', *Journal of Computer and System Sciences* **60**(1), 1–12.
URL: *https://doi.org/10.1006/jcss.1999.1661*

Magnanti, T. L. & Wong, R. T. (1984), 'Network design and transportation planning: Models and algorithms', *Transportation science* **18**(1), 1–55.
URL: *https://doi.org/10.1287/trsc.18.1.1*

Manurangsi, P. & Moshkovitz, D. (2017), 'Improved approximation algorithms for projection games', *Algorithmica* **77**(2), 555–594.
URL: *https://doi.org/10.1007/s00453-015-0088-5*

Molenbruch, Y., Braekers, K. & Caris, A. (2017), 'Typology and literature review for dial-a-ride problems', *Annals of Operations Research* **259**(1), 295–325.
URL: *https://doi.org/10.1007/s10479-017-2525-0*

Papadimitriou, C. H. & Yannakakis, M. (1982), The complexity of facets (and some facets of complexity), *in* 'Proceedings of the fourteenth annual ACM symposium on Theory of computing', pp. 255–260.

Papadimitriou, C. H. & Yannakakis, M. (1993), 'The traveling salesman problem with distances one and two', *Mathematics of Operations Research* **18**(1), 1–11.

Park, B. H., Seo, Y.-I., Hong, S.-P. & Rho, H.-L. (2013), 'Column generation approach to line planning with various halting patterns - application to the korean high-speed railway', *Asia-Pacific Journal of Operational Research* **30**(04), 1350006.
URL: *https://doi.org/10.1142/s0217595913500061*

Parragh, S. N., Pinho de Sousa, J. & Almada-Lobo, B. (2015), 'The dial-a-ride problem with split requests and profits', *Transportation Science* **49**(2), 311–334.
URL: *https://doi.org/10.1287/trsc.2014.0520*

Pei, J., Mladenović, N., Urošević, D., Brimberg, J. & Liu, X. (2020), 'Solving the traveling repairman problem with profits: A novel variable neighborhood search approach', *Information Sciences* **507**, 108–123.
URL: *https://doi.org/10.1016/j.ins.2019.08.017*

Peleg, D. (2007), 'Approximation algorithms for the label-covermax and red-blue set cover problems', *Journal of Discrete Algorithms* **5**(1), 55–64.
URL: *https://doi.org/10.1016/j.jda.2006.03.008*

Raghavan, P. & Tompson, C. D. (1987), 'Randomized rounding: a technique for provably good algorithms and algorithmic proofs', *Combinatorica* **7**(4), 365–374.

Schöbel, A. (2012), 'Line planning in public transportation: models and methods', *OR spectrum* **34**(3), 491–510.

Schöbel, A. & Scholl, S. (2006), Line planning with minimal traveling time, *in* '5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)', Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Seguin-Charbonneau, L. & Shepherd, F. B. (2021), 'Maximum edge-disjoint paths in planar graphs with congestion 2', *Mathematical Programming* **188**(1), 295–317.
URL: *https://doi.org/10.1007/s10107-020-01513-1*

Vansteenwegen, P., Souffriau, W. & Van Oudheusden, D. (2011), 'The orienteering problem: A survey', *European Journal of Operational Research* **209**(1), 1–10.

Williamson, D. P. & Shmoys, D. B. (2011), *The design of approximation algorithms*, Cambridge university press.

Wolsey, L. A. (1989), 'Submodularity and valid inequalities in capacitated fixed charge networks', *Operations Research Letters* **8**(3), 119–124.

Wong, R. T. (1980), 'Worst-case analysis of network design problem heuristics', *SIAM Journal on Algebraic Discrete Methods* **1**(1), 51–63.

Yu, J. & Dong, Y. (2013), 'Maximizing profit for vehicle routing under time and weight constraints', *International Journal of Production Economics* **145**(2), 573–583.
URL: *https://doi.org/10.1016/j.ijpe.2013.05.009*

국문초록

무회로 네트워크에서의 단일경로 최대이익다품종거래문제(sMTP)는 정해진 목적지로 가는 차량을 운행하여 마디 쌍 사이에 존재하는 운송 요청을 선택적으로 수행하여 이익을 최대화하는 문제이다. 무회로 유향 네트워크의 각 마디 쌍에는 한 마디에서 다른 마디로의 운송에 대한 요청이 존재한다. 시작 마디에서 마지막 마디까지의 경로가 정해지면, 그 경로 위의 각 요청의 출발지와 목적지 간의 물품을 수송량 제한을 넘지 않는 선에서 운송할 수 있다. 이 때 각 운송량에 비례한 수익을 얻는다. 같은 마디 쌍 사이의 운송 요청이라도 일부만 운송하여 일부의 수익만 얻을 수도 있다. 한 편, 운송을 하는 과정에서 단위 거리를 이동할 때 마다 수송량에 비례하는 비용이 발생한다. 따라서 수송량에 제한이 없더라도, 최대의 수익을 목적으로 모든 마디에 방문하는 것은 너무 멀리 우회를 하게 만들어 더 큰 비용을 초래할 수도 있다. 차량 운행자는 전체 마디 중 일부를 거치는 경로를 선택하고 각 운송 요청의 수행하여, 수익에서 비용을 뺀 이익을 최대화하는 것을 목적으로 한다.

본 논문에서는 sMTP에 대해, 가능해 집합에 대응하는 다면체의 구조와 근사 가능성에 대해 연구한다. 먼저, 정수 최적화 기반의 이론적 분석을 진행한다. sMTP를 위한 두 가지 모형을 제시한다. 각 모형에 대해 먼저 용량 제약이 없는 경우의 유효 부등식 집단들을 얻고, 각각이 강한 유효 부등식이 될 조건에 대해 논한다. 또한 발견한 유효 부등식들에 대응하는 분리 알고리듬을 제시한다. 다음으로, 해당 문제의 근사 가능성에 대한 분석을 진행한다. 먼저 근사해법 개발의 근본적 한계에 대해 다룬다. 계산 불가능성과 함께, 최악의 경우의 선형완화 문제의 해가 상한으로서 제공하는 품질의 한계에 대해 다룬다. 이 문제의 근사해법 개발을 위해, 몇몇 특수한 경우에 적용할 수 있는 근사해법을 제시한다. 제시한 해법들을 바탕으로 sMTP의 근사해법을 제시하고,

제시한 근사해법에 사용된 기법들이 활용될 수 있는 다른 문제들에 대해 다룬다. 마지막으로, 실험적으로 본 연구에서 제시한 유효 부등식을 이용한 다양한 해법의 성능과 근사해법을 활용한 해법의 성능을 비교한다.

주요어: 화물 운송 문제, 정수 최적화, 유효 부등식, 근사 불능성, 근사해법

학번: 2017-20407