TOPICS IN DISCRETE OPTIMIZATION: MODELS, COMPLEXITY AND ALGORITHMS

A Thesis
Presented to
The Academic Faculty

by

Qie He

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2013

TOPICS IN DISCRETE OPTIMIZATION: MODELS, COMPLEXITY AND ALGORITHMS

Approved by:

Dr. Shabbir Ahmed, Co-advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. George L. Nemhauser, Co-advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. William J. Cook
Department of Combinatorics and Optimization
University of Waterloo

Dr. Santanu S. Dey
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Robin Thomas
School of Mathematics
Georgia Institute of Technology

Date Approved: July 30, 2013

To my parents.

ACKNOWLEDGEMENTS

Many have told me that the most important piece of a dissertation is the acknowledgment part (not including my advisers). I cannot agree with them more. So...

I am extremely thankful to my advisers, George and Shabbir, for their constant support and guidance in every aspect of my graduate studies. Working with them continuously opens up my mind, and most importantly, helps me gain more convictions on what I love to do. Thanks also to Bill Cook, Santanu Dey and Robin Thomas for serving on my thesis committee.

I am deeply grateful for a lot of help received during my PhD studies. I want to thank Alex Shapiro and Hayriye Ayhan, for bearing my elementary questions on probability theory. Laurence Wolsey, for teaching me lot-sizing and always being so generous to answer my questions timely. Joel Sokol, for kindly letting me work with him in my first semester at Tech, which made the transition of studying in a new country much easier. Bill Cook, Robin Thomas, Hayriye Ayhan, Santanu Dey and Chen Zhou, for all the support and advices during my job search. Gary Parker, for being such a great program director, and the staff members in ISyE: Pam Morrison, Anita Race, Harry Sharp and Mark Reese, for making everything in the department much easier.

Research and learning become much more fun thanks to the discussions with fellow students in ISyE and the ACO program: Daniel Dadush, Xuefeng Gao, Cristobal Guzman, Fatma Kılınç-Karzan, Chun-hung Liu, Diego Moran, Dimitri Papageorgiou, Feng Qiu, Steve Tyber, Juan Pablo Vielma and so many more, and life certainly will not be so colorful without many graduate students in ISyE and a large community of Chinese friends I get to know in Atlanta.

Finally, I would like to thank my parents for standing me being a nerd and enjoying it. I want to thank my wife Shuhua for sharing every piece of bittersweet memory with me over the years. Half of the thesis is owing to her.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Discrete optimization problems arise in nearly every field of scientific and engineering interest. The pursuit of solutions for individual problem provides a better understanding of each problem's intrinsic complexity as well as the power and limits of the developed models of computation. On a more practical level, many discrete optimization problems are modeled and solved as mixed-integer programs. The art of modeling and the development of general mixed-integer program (MIP) solvers have great influence on finding satisfying solutions efficiently in practice. The prevalent presence of dynamics and uncertainties imposes even greater challenges on both tasks. In this thesis, we examine several discrete optimization problems through the perspectives of modeling, complexity and algorithms.

Cutting planes play a central role in the theory and computation of MIP. In Chapter 2, we propose the first probabilistic model to compare the strength of the traditional split cut and one type of newly developed two-row cut. We consider a two-row MIP with two integer variables, which is essentially a relaxation of the general MIP. The model is of particular interest since some of the facet-defining inequalities of the integer hull are not the traditional split cuts. It turns out that each non-trivial facet-defining inequality of the integer hull of this relaxation can be obtained from either a split, a triangle or a quadrilateral. These cuts are called split cuts, triangle cuts and quadrilateral cuts, respectively. It has been shown that the triangle closure and quadrilateral closure is strictly contained in the split closure, respectively, but in computation there is no clear evidence that these triangle or quadrilateral cuts always perform better than split cuts (sometimes even worse). To understand the mismatch between the theoretical strength and computational effectiveness of the new families of cuts, we propose a probabilistic model to compare the strength of the split cut and one type of triangle cut. Specifically, we address the following question: what is the likelihood that a split cut will dominate with respect to cut coefficients or cut off more volume from the linear programming (LP) relaxation than a type 1 triangle cut

for an arbitrary instance of the two-row MIP given a specific probability distribution of the problem parameters? Our analysis reveals that, for the given distribution of the instances, such likelihood is high. The analysis also suggests some guidelines on when type 1 triangle cuts are likely to be more effective than split cuts and vice versa.

In Chapter 3, we study a minimum concave cost network flow problem over a grid network (CFG). A grid network has vertices corresponding to a two-dimensional square lattice and horizontal and vertical arcs. In many applications, one dimension of the network models the temporal dynamics (time periods) and the other models the spatial locations (echelons). Concave cost functions are used to model economies of scale or a cost structure with a fixed setup cost. CFG models many practical problems in green recycling, production planning and transportation. We assume that the concave cost function is given by a function-value oracle for each arc. We give a polynomial-time algorithm to solve this problem when the number of echelons is fixed. We show that the problem is NP-hard when the number of echelons is an input parameter. We also extend our result to CFG with backward and upward arcs, which models backlogging and return of products respectively in supply chain management. Our result unifies the complexity results for the lot-sizing problem and several variants (multi-echelon, backlogging) in production planning and the pure remanufacturing problem in green recycling, and gives the first polynomial-time algorithm for some problems whose complexities were not known before. In addition, our technique based on path decomposition of extreme flows provides a unified framework to analyze the complexity of various lot-sizing models.

Finally, in Chapter 4, we examine how much complexity randomness will bring to a simple combinatorial optimization problem. We study a problem called the sell or hold problem (SHP). SHP is to sell k out of n indivisible assets over two stages, with known first-stage prices and random second-stage prices, to maximize the total expected revenue. SHP can be essentially formulated as a two-stage stochastic program with first-stage binary decision variables and second-stage continuous recourse variables. Although the deterministic version of SHP is trivial to solve, we show that SHP is NP-hard when the second-stage prices are realized as a finite set of scenarios. We show that SHP is polynomially solvable when the number of scenarios in the second stage is constant. A $\max\{1/2, k/n\}$-approximation algorithm is presented for the scenario-based SHP.

CHAPTER I

INTRODUCTION

The scope of this thesis is to investigate several discrete optimization problems through the perspectives of modeling, complexity and algorithms. Discrete optimization problems search for a best solution under certain criteria among a finite or countable set of feasible solutions. When the feasible solutions possess additional combinatorial structures, mostly related to graphs and set systems, these problems are also called combinatorial optimization problems. Combinatorial optimization arises in nearly every field of scientific and engineering interest, including many well-known problems such as the min cut problem, matching, the knapsack problem and the traveling salesman problem (TSP). The most important open question in combinatorial optimization is the infamous P=NP conjecture, asking whether a decision problem for which each "yes" instance has a certificate that can be verified in polynomial time can also be solved in polynomial time. The persistent absence of a positive answer inspires people to resort to various approximation algorithms for NP-hard problems [96]. Beyond the theoretical interest, a practical way to solve these problems is to formulate them as integer programming (IP) problems, and use a standard "solver" for the general IP models to find an optimal solution. It is not surprising that no solver in practice is computationally efficient for every input IP model. To help improve the performance of the solver for a particular problem, one usually needs to heavily exploit the combinatorial structure of the problem to strengthen the IP formulation. The intertwinement of combinatorial optimization and IP spawns the field of polyhedral combinatorics [32, 89].

As a generic modeling framework, IP has been a vigorous area since its inception in the 1950s. The theoretical and computational studies of IP have been advanced greatly in the past five decades [88, 82, 102, 74]. One method to solve a general IP model is the ingenious cutting-plane method, which was proposed by Ralph Gomory in 1958 [62, 63] and shown to terminate in a finite number of steps. Despite its theoretical elegance, the computational efficacy of the cutting-plane method is best exploited by incorporating it in a branch-and-cut framework. Probably the most challenging question related to the cutting-plane method is: how can we generate *strong* cuts in an *efficient* way? There has been extensive studies revolving around this question from both theoretical and practical perspectives, such as efficient ways to generate cuts (including the Chvátal-Gomory cut [62, 30], Gomory mixed integer (GMI) cut [63], Mixed integer rounding (MIR) cut [83], and lift-and-project cut [9, 10]), characterization of the minimal and extremal valid functions from the group relaxation [59, 60, 61], strengthening cuts by additional information [11, 100, 101, 64, 65, 4] and the comparisons of various cut families [33, 39]. Note that in this thesis the terms “cuts”, “cutting planes” and “valid inequalities” are used interchangeably, and “valid functions” are used for different types of infinite relaxations for the IP model. Although there are many ways to generate cuts for general IP models, it has been shown that the generic cuts currently used in commercial solvers, such as CPLEX, GUROBI and XPRESS, are essentially equivalent in terms of so-called elementary cut closure [33]. Furthermore they can all be seen as cuts derived from certain split disjunctions [31], and generated from a 2-slope one-dimensional valid function [61]. However for some IP instances, there are valid inequalities that can not be generated in such a way [31, 78]. It was not until recently that people have had a deeper understanding of these new families of cuts. The major breakthrough is due to a rediscovery of the one-to-one correspondence between the cut (valid function) and the lattice-free convex set (which does not contain any integer point in its interior) for some simple-structured IP model [3]. This nice geometric characterization provides a way to study the strength of the cuts in term of the corresponding lattice-free convex set. In particular, the maximal lattice-free convex set with nonempty interior in $\mathbb{R}^2$ is well understood and can only be a split, a triangle or a quadrilateral [79]. There have been extensive studies on conditions on when these cuts are facet-defining [34], the closure and rank comparison of different cut families [13, 45, 48], how to strengthen these cuts with additional information such as integrality constraints on some variables, non-negativity and other linear constraints [50, 51, 15, 58] and computational experiments on the new families of cuts [54, 13, 47]. Inspired by the results in $\mathbb{R}^2$, several groups have also carried ongoing research on deriving cuts from lattice-free convex set in higher dimensions [27, 16, 7], cuts from more general disjunctions such as a cross disjunction, a crooked cross disjunction [37, 40] and a multi-branch split disjunction [78, 38], generalized intersection cut [12], and strong valid functions for two and higher dimensional infinite group relaxations [49, 35, 20, 18, 19]. To understand how strong these new cuts are, we propose in Chapter 2 the first probabilistic model to compare the strength of the traditional split cuts and one type of newly developed two-row cuts.

Multistage decision making is a central topic in operations research and management science. A few combinatorial optimization problems, such as lot-sizing, inventory control and dynamic pricing, are cast in the fashion of planning or allocating limited resources over a number of stages, thereby naturally fall into this category. Many others, despite the lack of an explicit concept of "stage" in the problem statement, can be recast as multistage decision making problems, such as the shortest path problem, the knapsack problem and TSP. Besides the aforementioned IP model (or more generally the mathematical programming (MP) model), dynamic programming (DP), proposed by Richard Bellman [21], is another powerful modeling tool and solution approach for multistage decision making. Under the deterministic and discrete-time setting where the problem data at each stage are known to be certain, the MP and DP models should agree with each other, in that the optimal solutions obtained by solving each model should be the same under the same criteria. Meanwhile, due to the different emphasis from the modeling perspective, independent theory and solution methods have been developed for MP and DP, which in turn enjoy their own generality and limitations. Deterministic MP models the whole problem in a static way, and exploits time dynamics during the search for optimal solutions (such as decomposition of the model through stages). The design of efficient algorithms for general MP models usually relies on the existence of a duality theory, or a certificate that can be used efficiently to check the optimality of obtained solutions. On the other hand, DP is built upon a dynamic system with properly chosen state and action variables, and the optimal "policy" is characterized by the Bellman equation. Despite its theoretical modeling power, DP suffers from a well-known phenomenon called the *curse of dimensionality*, which states that the running time of solving the Bellman equation grows exponentially in the dimension of the state space, rendering DP intractable for many problems in practice. Connections between the two models have been explored in order to overcome the the limitations of one model by characteristics of the other. (1) Use DP as a modeling tool, approximate the optimal value function of DP by a class of prescribed functions with simple structure, and then solve the Bellman equation by a large-sized linear programming [41, 42]. (2) Derive extended formulations of the MP formulation from a DP algorithm [81, 80], which is used in turn to show the tightness of various MP formulations. This idea is applied extensively in the context of the lot-sizing problems [84] and more general fixed-charge network flow problems [86]. (3) Use DP within a branch-and-bound framework to solve the MP model. DP can either serve as a heuristic to solve subproblems of small sizes in the branch-and-bound tree, or provide a lower bound for the MP formulation by relaxing the state space [29, 91]. In Chapter 3, we explore a connection between MP and DP models to derive an efficient algorithm for the minimum concave cost flow problem over a grid network. The main idea is to exploit the algebraic and combinatorial structure of the optimal solution from the MP formulation to alleviate the curse of dimensionality for the DP formulation.

Although MP and DP reach the same end under the deterministic setting, various multistage decision making models diverge when uncertainty is taken into account. The presence of uncertainty elicits different perspectives on how uncertainty is modeled and quantified, how the dynamics of decision making interact with the uncertainty, and how the "optimal" solution (decision, policy) is defined. Following these perspectives, a few models have been proposed, including multistage stochastic programs [26, 90], Markov decision processes [85, 24], multistage robust optimization [22, 25] and stochastic optimal control [23], with various degrees of tractability and generality. In this thesis, we are particularly interested in the multistage stochastic integer programs (SIP), where uncertainty is modeled as a discrete-time stochastic process $\{\xi_t\}_{t=1}^T$ with ξ_t being a random vector whose realization is revealed at stage t. Decisions are made over T stages with certain components of the decision vector required to be integers. The objective is to optimize some risk functional such as expectation or conditional value at risk. The difficulty of SIP is multifold: the evaluation of the objective functional usually requires the calculation of a multi-dimensional integral, which is computational intractable in general; the number of decision variables grows exponentially in the number of stages; the presence of integer decision variables brings more nonlinearity and non-convexity into the model. One condition that could possibly lead to tractable SIP is that ξ_t only has a finite support for each stage t. Then the uncertainty information structure can be interpreted as a scenario tree, and SIP can be reduced to an equivalent deterministic IP, which is called the extensive form of the SIP. Motivated to examine the complexity of this particular SIP model, we study in Chapter 4 a two-stage stochastic combinatorial problem with a finite number of scenarios and a simple cardinality constraint, for which the deterministic version of the problem is trivial.

As discussed above, we pursue the three perspectives of discrete optimization problems in this thesis. Our main contributions are:

1. In Chapter 2, we propose the first probabilistic model to compare the strength of the traditional split cut and one type of newly developed two-row cut. We consider a two-row mixed-integer program (MIP) with two integer variables, which is essentially a relaxation of the general MIP. Then any valid inequality for this relaxation will also be valid for the general MIP. The model is of particular interest since some of the facet-defining inequalities of the integer hull are not the traditional split cuts. It turns out that each non-trivial facet-defining inequality of the integer hull of this relaxation can be obtained from either a split, a triangle or a quadrilateral. These cuts are called split cuts, triangle cuts and quadrilateral cuts, respectively. It has been shown that the triangle closure and quadrilateral closure is strictly contained in the split closure, respectively, but in computation there is no clear evidence that these triangle or quadrilateral cuts always perform better than split cuts (sometimes even worse). To understand the mismatch between the theoretical strength and computational effectiveness of the new families of cuts, we propose a probabilistic model to compare the strength of the split cut and one type of triangle cut. Specifically, we address the following question: what is the likelihood that a split cut will dominate with respect to cut coefficients or cut off more volume from the linear programming (LP) relaxation

than a type 1 triangle cut for an arbitrary instance of the two-row MIP given a specific probability distribution of the problem parameters? Our analysis reveals that, for the given distribution of the instances, such likelihood is high. The analysis also suggests some guidelines on when type 1 triangle cuts are likely to be more effective than split cuts and vice versa.

2. In Chapter 3, we study a minimum concave cost network flow problem over a grid network (CFG). A grid network has vertices corresponding to a two-dimensional square lattice and horizontal and vertical arcs. In many applications, one dimension of the network models the temporal dynamics (time periods) and the other models the spatial locations (echelons). Concave cost functions are used to model economies of scale or a cost structure with a fixed setup cost. CFG models many practical problems in green recycling, production planning and transportation. We assume that the concave cost function is given by a function-value oracle for each arc. We give a polynomial-time algorithm to solve this problem when the number of echelons is fixed. We show that the problem is NP-hard when the number of echelons is an input parameter. We also extend our result to CFG with backward and upward arcs, which models backlogging and return of products respectively in supply chain management. Our result unifies the complexity results for the lot-sizing problem and several variants (multi-echelon, backlogging) in production planning and the pure remanufacturing problem in green recycling, and gives the first polynomial-time algorithm for some problems whose complexities were not known before. In addition, our technique based on path decomposition of extreme flows provides a unified framework to analyze the complexity of various lot-sizing models.

3. In Chapter 4, we are interested in how much complexity randomness will bring to a simple combinatorial optimization problem. We study a problem called the sell or hold problem (SHP). SHP is to sell k out of n indivisible assets over two stages, with known first-stage prices and random second-stage prices, to maximize the total expected revenue. SHP can be essentially formulated as a two-stage stochastic program with

first-stage binary decision variables and second-stage continuous recourse variables. Although the deterministic version of SHP is trivial to solve, we show that SHP is NP-hard when the second-stage prices are realized as a finite set of scenarios. We show that SHP is polynomially solvable when the number of scenarios in the second stage is constant. A $\max\{1/2, k/n\}$-approximation algorithm is presented for the scenario-based SHP.

CHAPTER II

A PROBABILISTIC COMPARISON OF SPLIT AND TYPE 1 TRIANGLE CUTS FOR TWO ROW MIXED-INTEGER PROGRAMS

2.1 Introduction

This chapter is concerned with valid inequalities for a two-row mixed-integer program (MIP) with two integer variables of the form

$$\begin{aligned} & x = f + \sum_{j=1}^{k} r^j y_j \\ & x \in \mathbb{Z}^2,\ y_j \geq 0, \end{aligned} \tag{1}$$

where $f \in \mathbb{Q}^2 \setminus \mathbb{Z}^2$ and $r^j \in \mathbb{Q}^2 \setminus \{0\}$ for all j. Let X denote the set of solutions to (1). It has been shown (e.g., Andersen et al. [3]) that any valid inequality for $\text{conv}(X)$ that cuts off the infeasible point $(x, y) = (f, 0)$ is an intersection cut (Balas [8]), corresponding to a convex set $L \in \mathbb{R}^2$ with $\text{int}(L) \cap \mathbb{Z}^2 = \emptyset$ (i.e., integer-free or lattice-free) and $f \in \text{int}(L)$. Such a cut is of the form

$$\sum_{j=1}^{k} \psi_L(r^j) y_j \geq 1\,, \tag{2}$$

where $\psi_L : \mathbb{Q}^2 \mapsto \mathbb{R}$ is given by

$$\psi_L(r) = \begin{cases} 0 & r \in \text{rec.cone}(L) \\ \frac{1}{\lambda} & \lambda > 0,\ f + \lambda r \in \text{boundary}(L). \end{cases} \tag{3}$$

Furthermore, minimal inequalities of the form (2) can be derived from maximal integer-free sets in $\mathbb{R}^2$ with non-empty interior. As shown in Figure 1, such sets are of one of the following types (Lovász [79]):

- A split S: $c \leq ax_1 + bx_2 \leq c + 1$, where $a, b, c \in \mathbb{Z}$ and $gcd(a, b) = 1$;
- A triangle with an integer point in the relative interior of each of the edges; these can be further classified in to one of the following three types (Dey and Wolsey [50]):

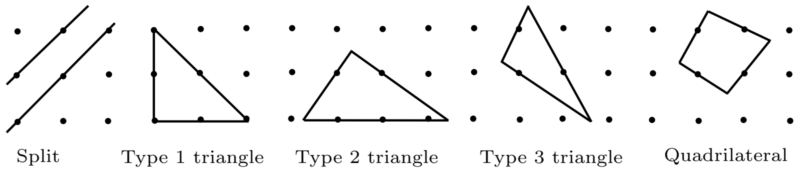


Figure 1: The integer-free bodies in $\mathbb{R}^2$ with non-empty interior

  - A type 1 triangle T_1: a triangle with integer vertices and exactly one integer point in the relative interior of each edge.
  - A type 2 triangle T_2: a triangle with more than one integer point on one edge and exactly one integer point in the relative interior of each of the other two edges.
  - A type 3 triangle T_3: a triangle with exactly one integer point in the relative interior of each edge and non-integral vertices.

- A quadrilateral Q with exactly one integer point in the relative interior of each edge such that the four integer points form a parallelogram of area one.

Inequalities of the form (2) corresponding to the above sets are called split, (type 1, 2 or 3) triangle, and quadrilateral cuts, respectively. From the maximality of the above integer-free sets, it follows that any non-trivial facet-defining inequality of $\text{conv}(X)$ is either a split, triangle or quadrilateral cut [3, 34].

Split cuts are the classical GMI or MIR cuts [82]. Recently there has been a great deal of activity in comparing triangle and quadrilateral cuts to split cuts for two-row MIPs. Basu et al. [14] compared the rank-1 closure (the convex set obtained by adding in a single round all possible cuts from the family) corresponding to the three cuts classes. They showed that the triangle closure (considering all three types of triangle cuts) and the quadrilateral closure are contained in the split closure, suggesting that triangle and quadrilateral cuts are in some sense stronger than split cuts. Dey [48] showed that type 2, type 3 triangle cuts and quadrilateral cuts have a finite split ranks (i.e., such a cut can be constructed via a finite sequence of split cuts) while only type 1 triangle cuts can have infinite split rank. However, empirical studies demonstrating the success of triangle and quadrilateral cuts in comparison to split (or GMI) cuts have been limited. Espinoza [54] reported some success with intersection cuts generated from some classes of integer-free triangles and quadrilaterals. Basu et al. [13] considered strengthened versions of a class of type 2 triangle cuts and showed that combining these cuts with GMI cuts give somewhat better performance than GMI cuts alone. Dey et al. [46] presented computational results on randomly generated multi-knapsack instances and showed that a subclass of type 2 triangle cuts can close more gap than GMI cuts.

We present a probabilistic comparison of type 1 triangle cuts and split cuts. Specifically we address the question: what is the likelihood that a split cut will dominate with respect to cut coefficients or cut off more volume from the linear programming relaxation than a type 1 triangle cut for an arbitrary instance of the two-row MIP (1) given a specific probability distribution of the problem parameters? Our analysis reveals that, for the given distribution of the instances, such likelihood is high. The analysis also suggests some guidelines on when type 1 triangle cuts are likely to be more effective than split cuts and vice versa. The result in this chapter is a joint work with Shabbir Ahmed and George Nemhauser and appeared in [71].

2.2 Setup

In this section, we discuss the distributional model for instances of the two-row MIP (1) and the two metrics used in our probabilistic comparison of type 1 triangle and split cuts.

Without loss of generality, (by translating x by $\lfloor f \rfloor$ and scaling y_j by $\|r^j\|_2$) we can assume that $0 \leq f_i < 1$ for $i = 1, 2$ and $\|r^j\|_2 = 1$ for all j in (1). Then $r_1^j = \cos\theta_j$ and $r_2^j = \sin\theta_j$ where θ_j is the angle between r^j and the positive x_1-axis.

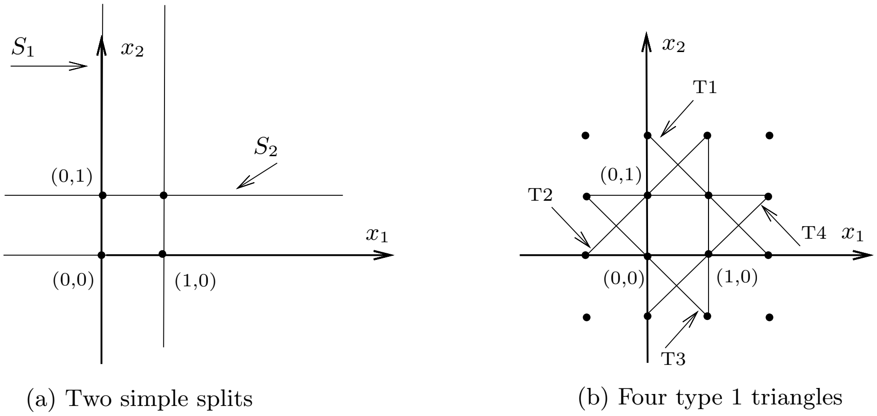
The input model: We consider instances of (1) where f is a realization of a random vector $\mathbf{f}$ that is uniformly distributed with support $U := (0,1)^2$, i.e., the open unit square in the plane, and θ_j is a realization of a random variable $\boldsymbol{\theta_j}$ that is uniformly distributed over $[0, 2\pi)$ for all j. (When f is on the boundary of $\text{cl}(U)$, the coefficients for some split and type 1 triangle cuts can be $+\infty$, causing technical issues in their comparison.) Moreover,

$\mathbf{f}, \boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_k}$ are independent random variables.

Under this probabilistic input model, the cut corresponding to the integer-free body L is of the form

$$\sum_{j=1}^{k} \psi_L(\mathbf{f}, \boldsymbol{\theta_j}) y_j \geq 1, \tag{4}$$

where the cut coefficient $\psi_L(\mathbf{f}, \boldsymbol{\theta_j})$ of variable y_j is a random variable depending on $\mathbf{f}$ and $\boldsymbol{\theta_j}$ and is given by (3). Our analysis compares the random cut (4) when the set L is a split or a type 1 triangle. To guarantee that $f \in \text{int}(L)$ with probability one, we only consider integer-free splits and type 1 triangles that contain U. This ensures that the inequality (4) corresponding to L cuts off the infeasible point $(f, 0)$ for every realization f of $\mathbf{f}$. There are only two splits containing U (the valid inequality corresponds to the GMI cut for each row of system (1)) and there are only four type 1 triangles containing U, with one of the four vertices of U as its right-angle vertex (see Figure 2).



(a) Two simple splits

(b) Four type 1 triangles

Figure 2: The integer-free bodies selected for comparison

There are various criteria for comparing cuts. We choose two criteria suitable for comparing two individual cuts rather than cut families. The first one is based on cut dominance.

Definition 1. Suppose $C_1 : \sum_{j=1}^{k} a_j y_j \geq 1$ and $C_2 : \sum_{j=1}^{k} b_j y_j \geq 1$ are two distinct valid inequalities for system (1), then C_1 **dominates** C_2 if $a_j \leq b_j$ for $j = 1, \cdots, k$ with at least one of the inequalities being strict. We use $C_1 \succ_D C_2$ to denote that C_1 dominates C_2.

If $C_1 \succ_D C_2$, then C_2 is implied by C_1. The second criteria is based on the volume cut off by the cuts from the linear relaxation.

Definition 2. Suppose $C_1 : \sum_{j=1}^{k} a_j y_j \geq 1$ and $C_2 : \sum_{j=1}^{k} b_j y_j \geq 1$ are two distinct valid inequalities for system (1). Let X_{LP} be the linear relaxation of (1). Then $C_1 \succ_V C_2$ if C_1 **cuts off more volume** than C_2 from X_{LP}, i.e.

$$\mathrm{vol}(X_{LP} \cap \{(x,y) : \sum_{j=1}^{k} a_j y_j \leq 1\}) > \mathrm{vol}(X_{LP} \cap \{(x,y) : \sum_{j=1}^{k} b_j y_j \leq 1\}).$$

We probabilistically compare split and type 1 triangle cuts with respect to these two metrics.

2.3 Conditional Probabilities with respect to f

We first analyze the conditional probabilities of split cuts dominating and cutting off more volume than triangle cuts with respect to the fractional point f. This analysis helps with computing the total probabilities in Section 2.4, and also provides some insight into values of f for which type 1 triangle cuts are likely to be better than split cuts and vice versa.

2.3.1 Cut coefficient comparison

Without loss of generality, we select one split from the two splits and one type 1 triangle from the four type 1 triangles in Figure 2. The analysis easily extends to the other splits and type 1 triangles by symmetry. The chosen split S_1 and type 1 triangle T_1 are shown in Figure 3. The split S_1 is defined by AD and BC and the type 1 triangle T_1 is defined by $\triangle AEF$. Suppose that C_{S_1} is the split cut for S_1 and C_{T_1} is the triangle cut for T_1, and recall that $\psi_{S_1}(\mathbf{f}, \boldsymbol{\theta_j})$ and $\psi_{T_1}(\mathbf{f}, \boldsymbol{\theta_j})$ are the corresponding (random) cut coefficients for variable y_j. We use $\Pr[\psi_{\mathrm{T}_1}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{S}_1}(\mathbf{f}, \boldsymbol{\theta_j}) | \mathrm{f}]$ to denote the conditional probability of the event $\psi_{T_1}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{S_1}(\mathbf{f}, \boldsymbol{\theta_j})$ when $\mathbf{f} = f$.

Lemma 1. *For each $j = 1, \cdots, k$, $\Pr[\psi_{\mathrm{T}_1}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{S}_1}(\mathbf{f}, \boldsymbol{\theta_j}) | \mathrm{f}] = \alpha(\mathrm{f})$, $\Pr[\psi_{\mathrm{S}_1}(\mathbf{f}, \boldsymbol{\theta_j}) = \psi_{\mathrm{T}_1}(\mathbf{f}, \boldsymbol{\theta_j}) | \mathrm{f}] = \beta(\mathrm{f})$ and $\Pr[\psi_{\mathrm{S}_1}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{T}_1}(\mathbf{f}, \boldsymbol{\theta_j}) | \mathrm{f}] = \gamma(\mathrm{f})$, where*

$$\alpha(f) = \frac{\arccos \frac{f_2(f_2-1)+(1-f_1)^2}{\sqrt{[f_2^2+(1-f_1)^2][(1-f_2)^2+(1-f_1)^2]}}}{2\pi}, \qquad \beta(f) = \frac{\arccos \frac{f_1^2+f_2^2-2f_2}{\sqrt{[f_1^2+f_2^2][f_1^2+(2-f_2)^2]}}}{2\pi},$$

$$\gamma(f) = \frac{\arccos \frac{f_1^2+f_2^2-f_1}{\sqrt{[f_1^2+f_2^2][(1-f_1)^2+f_2^2]}} + \arccos \frac{f_1^2+f_2^2-f_1-3f_2+2}{\sqrt{[(1-f_2)^2+(1-f_1)^2][f_1^2+(2-f_2)^2]}}}{2\pi}$$

Proof. Since $\boldsymbol{\theta_j}$ $(j = 1, \cdots, k)$ are i.i.d., we only need to prove the result for some j. For simplicity, we supress the index j here and prove it for some ray $r = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$.

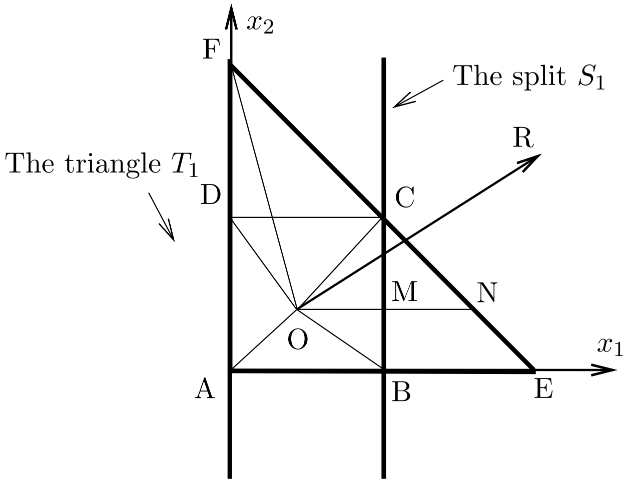


Figure 3: Computing $\Pr[\psi_S(\mathbf{f}, \boldsymbol{\theta}) < \psi_{T_1}(\mathbf{f}, \boldsymbol{\theta})]$

As shown in Figure 3, U is the unit square with vertices A, B, C and D and O is the fractional point f. Let OR be the ray defined by $f + \lambda r$. Let OM be the line parallel to the x_1-axis that intersects S and T_1 at M and N respectively. Then θ is the angle between OM and OR in the counterclockwise direction. Let the symbol $\angle$ denote an angle less than π. Since the probability density function of $\boldsymbol{\theta}$ is $\frac{1}{2\pi} I(\theta \in [0, 2\pi))$, by the law of total probability,

$$\Pr[\psi_{\mathrm{S}_1}(\mathbf{f}, \boldsymbol{\theta}) < \psi_{\mathrm{T}_1}(\mathbf{f}, \boldsymbol{\theta}) | \mathrm{f}] = \int_0^{2\pi} \frac{\mathbb{I}(\psi_{\mathrm{S}_1}(\mathrm{f}, \theta) < \psi_{\mathrm{T}_1}(\mathrm{f}, \theta))}{2\pi} \mathrm{d}\theta, \tag{5}$$

where $\mathbb{I}(A)$ is the indicator function of event A.

By (3), $\psi_{S_1}(f, \theta) = \frac{1}{\lambda_{S_1}}$, where $f + \lambda_{S_1} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \in \text{boundary}(S)$, and $\psi_{T_1}(f, \theta) = \frac{1}{\lambda_{T_1}}$ where $f + \lambda_{T_1} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \in \text{boundary}(T_1)$. Therefore, $\psi_{S_1}(f, \theta) < \psi_{T_1}(f, \theta)$ if the ray $f + \lambda \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$ hits the boundary of T_1 first, and $\psi_{T_1}(f, \theta) < \psi_{S_1}(f, \theta)$ if the ray $f + \lambda \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$ hits the boundary of S_1 first. When $\theta \in [0, \angle MOC)$ or $\theta \in (2\pi - \angle MOB, 2\pi)$,

OR is contained in the cone bounded by OB and OC, and hits the boundary of S first, so $\psi_{T_1}(f,\theta) < \psi_{S_1}(f,\theta)$. Similarly, when $\theta \in (\angle MOC, \angle MOF)$ or $\theta \in (2\pi - \angle MOA, 2\pi - \angle MOB)$, $\psi_{S_1}(f,\theta) < \psi_{T_1}(f,\theta)$; when $\theta \in [\angle MOF, 2\pi - \angle MOA]$ or θ is equal to $\angle MOC$ or $2\pi - \angle MOB$, $\psi_{S_1}(f,\theta) = \psi_{T_1}(f,\theta)$. Therefore, by (5),

$$\Pr[\psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta}) < \psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \frac{\angle \mathrm{AOB} + \angle \mathrm{COF}}{2\pi}, \quad \Pr[\psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta}) = \psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \frac{\angle \mathrm{AOF}}{2\pi},$$

$$\Pr[\psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta}) < \psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \frac{\angle \mathrm{BOC}}{2\pi}.$$

In $\triangle BOC$, $|\overline{OB}| = \sqrt{(1-f_1)^2 + f_2^2}$, $|\overline{OC}| = \sqrt{(1-f_1)^2 + (1-f_2)^2}$ and $|\overline{BC}| = 1$. By the law of cosines,

$$\begin{aligned}\cos \angle BOC &= \frac{|\overline{OB}|^2 + |\overline{OC}|^2 - |\overline{BC}|^2}{2|\overline{OB}||\overline{OC}|} \\ &= \frac{f_2(f_2-1) + (1-f_1)^2}{\sqrt{[f_2^2 + (1-f_1)^2][(1-f_2)^2 + (1-f_1)^2]}} = 2\pi\alpha(f).\end{aligned}$$

Therefore,

$$\Pr[\psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta}) < \psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \alpha(\mathrm{f}).$$

Similarly, $\angle AOF = 2\pi\beta(f)$ and $\angle AOB + \angle COF = 2\pi\gamma(f)$. Therefore,

$$\Pr[\psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta}) = \psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \beta(\mathrm{f}), \quad \Pr[\psi_{\mathrm{S}_1}(\mathbf{f},\boldsymbol{\theta}) < \psi_{\mathrm{T}_1}(\mathbf{f},\boldsymbol{\theta})|\mathrm{f}] = \gamma(\mathrm{f}).$$

□

Lemma 1 provides the probabilities that a single coefficient of the split cut C_{S_1} is smaller than, equal to, and larger than that of the triangle cut C_{T_1} as a function of f. To compare the other split and type 1 triangles in Figure 2, we only need to change f_1 to $1-f_1$ or f_2 to $1-f_2$ in $\alpha(f)$, $\beta(f)$ and $\gamma(f)$ by symmetry. The following theorem gives the conditional probability that the split cut C_{S_1} dominates the triangle cut C_{T_1} with respect to f and the number of continuous variables k.

Theorem 1.

$$\Pr[\mathrm{C}_{\mathrm{S}_1} \succ_{\mathrm{D}} \mathrm{C}_{\mathrm{T}_1}|\mathrm{f}] = [\beta(\mathrm{f}) + \gamma(\mathrm{f})]^{\mathrm{k}} - [\beta(\mathrm{f})]^{\mathrm{k}}.$$

Proof.

$$\begin{aligned}
&\mathrm{Pr}[\mathrm{C_{S_1}} \succ_{\mathrm{D}} \mathrm{C_{T_1}}|\mathrm{f}]\\
&= \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) \leq \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j}), \forall \mathrm{j}|\mathrm{f}] - \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) = \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j}), \forall \mathrm{j}|\mathrm{f}]\\
&= \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) \leq \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j})|\mathrm{f}]^{\mathrm{k}} - \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) = \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j})|\mathrm{f}]^{\mathrm{k}}\\
&= [\beta(f) + \gamma(f)]^k - [\beta(f)]^k,
\end{aligned}$$

where the second equality follows from the assumption that $\boldsymbol{\theta_j}$ $(j = 1, \cdots, k)$ are i.i.d..

□

Given integer-free bodies L_1 and L_2, let

$$R_D(L_1, L_2) = \{f \in U : \mathrm{Pr}[\mathrm{C_{L_1}} \succ_{\mathrm{D}} \mathrm{C_{L_2}}|\mathrm{f}] > \mathrm{Pr}[\mathrm{C_{L_2}} \succ_{\mathrm{D}} \mathrm{C_{L_1}}|\mathrm{f}]\}.$$

The following corollary follows from Theorem 1.

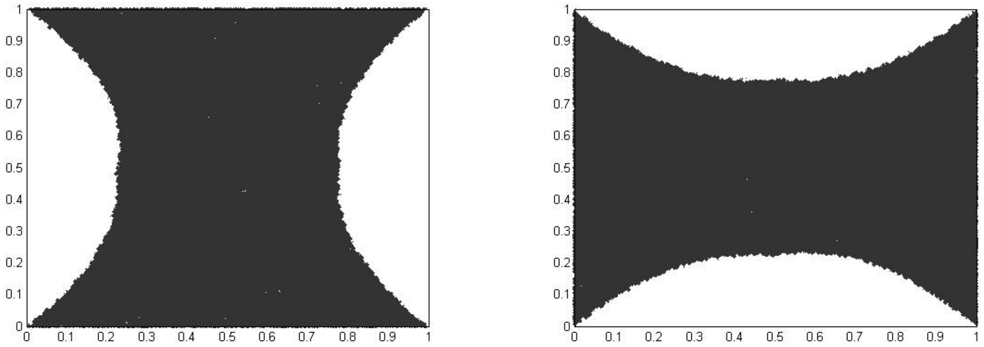
Corollary 1.

$$R_D(S_1, T_1) = \{f \in U : \gamma(f) > \alpha(f)\} \quad \textit{and} \quad R_D(T_1, S_1) = \{f \in U : \alpha(f) > \gamma(f)\}.$$

By symmetry, after appropriately translating f, we can similarly describe the regions $R_D(S_i, T_j)$ and $R_D(T_j, S_i)$ for $i = 1, 2$ and $j = 1, 2, 3, 4$ corresponding to any of the two splits and four type 1 triangles in Figure 2. Figures 4(a) and 4(b) show the regions $\cap_{j=1}^4 R_D(S_1, T_j)$ and $\cap_{j=1}^4 R_D(S_2, T_j)$, respectively shaded in black. The white regions in these figures indicate $\cup_{j=1}^4 R_D(T_j, S_1)$ and $\cup_{j=1}^4 R_D(T_j, S_2)$, respectively. Since the union of the two black regions covers the unit square, there is no f for which a type 1 triangle cut C_T satisfies that $\mathrm{Pr}[\mathrm{C_T} \succ_{\mathrm{D}} \mathrm{C_{S_i}}|\mathrm{f}] > \mathrm{Pr}[\mathrm{C_{S_i}} \succ_{\mathrm{D}} \mathrm{C_T}|\mathrm{f}]$ $(i = 1, 2)$. It follows from the discussion above that if we are only allowed to add one cut, when $f \in \cap_{j=1}^4 R_D(S_1, T_j)$, we would select S_1, and when $f \in \cup_{j=1}^4 R_D(T_j, S_1)$, we would select S_2.

2.3.2 Volume comparison

In this section, we compare cuts based on the volume cut off from the linear relaxation of system (1). First we describe how the volume cut off is computed.



(a) The region $\cap_{j=1}^{4} R_D(S_1, T_j)$

(b) The region $\cap_{j=1}^{4} R_D(S_2, T_j)$

Figure 4: The region.

Suppose that $C : \sum_{j=1}^{k} a_j y_j \geq 1$, with $a_j \geq 0$ for all j, is a valid inequality for system (1). Consider the linear relaxation of (1)

$$\begin{aligned} x &= f + \sum_{j=1}^{k} r^j y_j \\ x &\in \mathbb{R}^2,\ y_j \geq 0. \end{aligned} \tag{6}$$

Let X_{LP} be the set of feasible solutions of system (6) and

$$S_C = X_{LP} \cap \{(x, y) : \sum_{j=1}^{k} a_j y_j \leq 1\}.$$

Let vol(S_C) denote the volume of the polyhedron S_C, which is also the volume cut off from S by the valid inequality C. The following lemma gives the volume of S_C.

Lemma 2.

$$vol(S_C) = \begin{cases} +\infty & \textit{if } \exists j \textit{ such that } a_j = 0 \\ \frac{\alpha}{n! \prod_{j=1}^{k} a_j} & \textit{otherwise,} \end{cases} \tag{7}$$

where α is a constant depending on the rays $r^1, \cdots, r^k$.

Proof. When $a_j = 0$ for some j, S_C is an unbounded polyhedron, and vol$(S_C) = +\infty$. When $a_j > 0$ for all j, S_C is a k-dimensional polytope containing $(f, 0)$. Let

$$\text{Proj}_y(S_C) = \{y \in \mathbb{R}^k : \exists x \in \mathbb{R}^2 \text{ such that } (x, y) \in S_C\}$$

be the projection of S_C onto the y space. $\mathrm{Proj}_y(S_C)$ is a k-dimensional simplex with 0, $\frac{1}{a_1}e^1, \ldots, \frac{1}{a_k}e^k$ as its $(k+1)$ vertices, where e^j is the j-th unit vector. Therefore,

$$\mathrm{vol}(\mathrm{Proj}_y(S_C)) = \frac{1}{n!}\frac{1}{a_1}\cdots\frac{1}{a_k} = \frac{1}{n!\prod_{j=1}^{k} a_j}.$$

Each point in S_C is just an affine transformation of a point in the simplex $\mathrm{Proj}_y(S_C)$, so $\mathrm{vol}(S_C)$ and $\mathrm{vol}(\mathrm{Proj}_y(S_C))$ only differ by a factor α depending on the rays $r^1, \cdots, r^k$. Thus $\mathrm{vol}(S_C) = \frac{\alpha}{n!\prod_{j=1}^{k} a_j}$. □

By Lemma 2, it suffices to compute the product of cut coefficients when we compare cuts based on the volume cut off from the linear relaxation.

Now consider the split S_1 and type 1 triangle T_1 as in Section 2.3.1. As before, the analysis easily extends to another pair of split and type 1 triangle bodies by symmetry. Note that for fixed $f \in (0,1)^2$, $\psi_{T_1}(f, \boldsymbol{\theta_j}) > 0$ with probability one. Moreover, since $\boldsymbol{\theta_j}$ is continuously distributed, $\Pr[\exists \mathrm{j\ s.t.\ } \psi_{\mathrm{S_1}}(\mathrm{f}, \boldsymbol{\theta_j}) = 0] = \Pr[\exists \mathrm{j\ s.t.\ } \boldsymbol{\theta_j} = \frac{\pi}{2} \text{ or } \frac{3\pi}{2}] = 0$.

Theorem 2.

$$\Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}|\mathrm{f}] = \Pr[\sum_{\mathrm{j=1}}^{\mathrm{k}} \ln \frac{\psi_{\mathrm{S_1}}(\mathrm{f}, \boldsymbol{\theta_j})}{\psi_{\mathrm{T_1}}(\mathrm{f}, \boldsymbol{\theta_j})} < 0].$$

Proof. From Definition 2, Lemma 2 and the fact that $\psi_{S_1}(f, \boldsymbol{\theta_j}) > 0$ and $\psi_{T_1}(f, \boldsymbol{\theta_j}) > 0$ with probability one, we have that

$$\begin{aligned}\Pr[\mathrm{C_{S_1}} \succ_V C_{T_1}|f] &= \Pr[\mathrm{vol}(\mathrm{S_{C_{S_1}}}) > \mathrm{vol}(\mathrm{S_{C_{T_1}}})|\mathrm{f}] \\ &= \Pr[\frac{\alpha}{\mathrm{n!}\prod_{\mathrm{j=1}}^{\mathrm{k}} \psi_{\mathrm{S_1}}(\mathrm{f}, \boldsymbol{\theta_j})} > \frac{\alpha}{\mathrm{n!}\prod_{\mathrm{j=1}}^{\mathrm{k}} \psi_{\mathrm{T_1}}(\mathrm{f}, \boldsymbol{\theta_j})}] = \Pr[\sum_{\mathrm{j=1}}^{\mathrm{k}} \ln \frac{\psi_{\mathrm{S_1}}(\mathrm{f}, \boldsymbol{\theta_j})}{\psi_{\mathrm{T_1}}(\mathrm{f}, \boldsymbol{\theta_j})} < 0].\end{aligned}$$

□

Next we analyze the asymptotic behavior of the probability $\Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}|\mathrm{f}]$ as the number of continuous variables k increases. Before presenting further results, we give two technical lemmas.

Lemma 3.

$$\int_0^{\frac{\pi}{2}} \ln\cos x dx = -\frac{\pi \ln 2}{2} \quad \textit{and} \quad \int_0^{\frac{\pi}{2}} (\ln\cos x)^2 dx < \infty.$$

Proof. By substitution of variables, $\int_0^{\frac{\pi}{2}} \ln\cos x dx = \int_0^{\frac{\pi}{2}} \ln\sin x dx$. Then,

$$\begin{aligned}
\int_0^{\frac{\pi}{2}} \ln\sin x dx &= \int_0^{\frac{\pi}{2}} \ln(2\sin\frac{x}{2}\cos\frac{x}{2})dx \\
&= \int_0^{\frac{\pi}{2}} \ln 2 dx + \int_0^{\frac{\pi}{2}} \ln\sin\frac{x}{2}dx + \int_0^{\frac{\pi}{2}} \ln\cos\frac{x}{2}dx \\
&= \frac{\pi\ln 2}{2} + 2\int_0^{\frac{\pi}{4}} \ln\sin y dy + 2\int_0^{\frac{\pi}{4}} \ln\cos z dz \\
&= \frac{\pi\ln 2}{2} + 2\int_0^{\frac{\pi}{4}} \ln\sin y dy + 2\int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \ln\sin y dy \\
&= \frac{\pi\ln 2}{2} + 2\int_0^{\frac{\pi}{2}} \ln\sin y dy
\end{aligned}$$

Therefore, $\int_0^{\frac{\pi}{2}} \ln\sin x dx = -\frac{\pi\ln 2}{2}$.

By substitution of variables, $\int_0^{\frac{\pi}{2}} (\ln\cos x)^2 dx = \int_0^{\frac{\pi}{2}} (\ln\sin x)^2 dx$. Since $0 \le \sin x \le x$ for $0 \le x \le \frac{\pi}{2}$, then $0 \le (\ln\sin x)^2 \le (\ln x)^2$. Moreover, $\int (\ln x)^2 dx = x(\ln x)^2 - 2x\ln x + 2x + d$, where d is a constant. Thus, $\int_0^{\frac{\pi}{2}} (\ln x)^2 dx = \frac{\pi}{2}(\ln\frac{\pi}{2})^2 - \pi\ln\frac{\pi}{2} + \pi < \infty$. Therefore, $\int_0^{\frac{\pi}{2}} (\ln\sin x)^2 dx$ is finite. □

To simplify the notation, let $\boldsymbol{X_j}(f) = \ln\frac{\psi_{S_1}(f,\boldsymbol{\theta_j})}{\psi_{T_1}(f,\boldsymbol{\theta_j})}$ for every $j = 1, \dots, k$. Note that for a fixed $f \in (0,1)^2$, the random variable $\boldsymbol{X_j}(f)$ is uniquely determined by $\boldsymbol{\theta_j}$. The assumption that $\boldsymbol{\theta_j}$, for $j = 1, \dots, k$, are i.i.d. implies that $\boldsymbol{X_j}(f)$, for $j = 1, \dots, k$, are also i.i.d.. Let $\mu_f = \mathrm{E}[\boldsymbol{X_j}(f)]$ and $\sigma_f^2 = \mathrm{Var}[\boldsymbol{X_j}(f)]$ for any $j = 1, \dots, k$.

Lemma 4.

$$|\mu_f| < \infty \quad \textit{and} \quad \sigma_f^2 < \infty.$$

Proof. To simplify the notation, let

$$\overline{\boldsymbol{X}}_k(f) = \frac{\sum_{j=1}^k \boldsymbol{X_j}(f)}{k}, \qquad \mu_f = \mathrm{E}[\ln\psi_{S_1}(f,\boldsymbol{\theta_j})] - \mathrm{E}[\ln\psi_{T_1}(f,\boldsymbol{\theta_j})].$$

By (3), $\psi_{T_1}(f,\boldsymbol{\theta_j})$ is bounded and strictly positive for fixed $f \in (0,1)^2$. Thus $\ln\psi_{T_1}(f,\boldsymbol{\theta_j})$ is bounded and $\mathrm{E}[\ln\psi_{T_1}(f,\boldsymbol{\theta_j})]$ is finite. By (3), $\psi_{S_1}(f,\theta_j) = \frac{1}{\lambda_{S_1}}$ where $f + \lambda_{S_1}\begin{pmatrix}\cos\theta_j \\ \sin\theta_j\end{pmatrix}$ hits the boundary of the split S_1. Thus, $f_1 + \lambda_{S_1}\cos\theta_j = 1$ when $\theta_j \in [0, \frac{\pi}{2})$ and $\theta_j \in (\frac{3\pi}{2}, 2\pi)$, and $f_1 + \lambda_{S_1}\cos\theta_j = 0$ when $\theta_j \in (\frac{\pi}{2}, \frac{3\pi}{2})$. Therefore, $\psi_{S_1}(f,\theta_j) = \frac{\cos\theta_j}{1-f_1}$ when $\theta_j \in [0, \frac{\pi}{2})$

and $\theta_j \in (\frac{3\pi}{2}, 2\pi)$, and $\psi_{S_1}(f, \theta_j) = -\frac{\cos\theta_j}{f_1}$ when $\theta_j \in (\frac{\pi}{2}, \frac{3\pi}{2})$. The probability density function of $\boldsymbol{\theta_j}$ is $\frac{1}{2\pi} I(\theta_j \in [0, 2\pi))$. Therefore,

$$\begin{aligned}
&\mathrm{E}[\ln \psi_{S_1}(f, \boldsymbol{\theta_j})] = \int_0^{2\pi} \ln \psi_{S_1}(f, \theta_j) \frac{1}{2\pi} d\theta_j \\
&= \frac{1}{2\pi}[\int_0^{\frac{\pi}{2}} \ln \frac{\cos\theta_j}{1-f_1} d\theta_j + \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \ln \frac{-\cos\theta_j}{f_1} d\theta_j + \int_{\frac{3\pi}{2}}^{2\pi} \ln \frac{\cos\theta_j}{1-f_1} d\theta_j] \\
&= \frac{1}{2\pi}[\int_0^{\frac{\pi}{2}} \ln\cos\theta_j d\theta_j - \int_0^{\frac{\pi}{2}} \ln(1-f_1) d\theta_j + \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \ln(-\cos\theta_j) d\theta_j \\
&\quad - \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \ln f_1 d\theta_j + \int_{\frac{3\pi}{2}}^{2\pi} \ln\cos\theta_j d\theta_j - \int_{\frac{3\pi}{2}}^{2\pi} \ln(1-f_1) d\theta_j] \\
&= \frac{1}{2\pi}[4 \int_0^{\frac{\pi}{2}} \ln\cos\theta_j d\theta_j - \pi \ln f_1(1-f_1)]
\end{aligned}$$

By Lemma 3, $\int_0^{\frac{\pi}{2}} \ln\cos\theta_j d\theta_j = -\frac{\pi \ln 2}{2}$. Therefore, $\mathrm{E}[\ln \psi_{S_1}(f, \boldsymbol{\theta_j})]$ is finite and $\mu_f < \infty$. It only remains to verify that σ_f is finite. Since $\sigma_f^2 = \mathrm{E}[(\boldsymbol{X_j}(f))^2] - \mu_f^2$, we need to verify that $\mathrm{E}[(\boldsymbol{X_j}(f))^2]$ is finite.

$$\begin{aligned}
\mathrm{E}[(\boldsymbol{X_j}(f))^2] &= \mathrm{E}[(\ln \frac{\psi_{S_1}(f, \boldsymbol{\theta_j})}{\psi_{T_1}(f, \boldsymbol{\theta_j})})^2] \\
&= \mathrm{E}[(\ln \psi_{S_1}(f, \boldsymbol{\theta_j}))^2] - 2\mathrm{E}[\ln \psi_{S_1}(f, \boldsymbol{\theta_j}) \ln \psi_{T_1}(f, \boldsymbol{\theta_j})] + \mathrm{E}[(\ln \psi_{T_1}(f, \boldsymbol{\theta_j}))^2].
\end{aligned}$$

Since we have shown that $\ln \psi_{T_1}(f, \boldsymbol{\theta_j})$ is bounded and $\mathrm{E}[\ln \psi_{S_1}(f, \boldsymbol{\theta_j})]$ is finite for fixed f, the last two terms in the above equation are finite. For the first term $\mathrm{E}[(\ln \psi_{S_1}(f, \boldsymbol{\theta_j}))^2]$, substitute the formula for $\ln \psi_{S_1}(f, \theta_j)$ and expand it as an integration,

$$\begin{aligned}
&\mathrm{E}[(\ln \psi_{S_1}(f, \boldsymbol{\theta_j}))^2] \\
&= \int_0^{\frac{\pi}{2}} (\ln \frac{\cos\theta_j}{1-f_1})^2 \frac{1}{2\pi} d\theta_j + \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} (\ln \frac{-\cos\theta_j}{f_1})^2 \frac{1}{2\pi} d\theta_j + \int_{\frac{3\pi}{2}}^{2\pi} (\ln \frac{\cos\theta_j}{1-f_1})^2 \frac{1}{2\pi} d\theta_j \\
&= \frac{1}{2\pi}[4 \int_0^{\frac{\pi}{2}} (\ln\cos\theta_j)^2 d\theta_j - 4 \ln f_1(1-f_1) \int_0^{\frac{\pi}{2}} \ln\cos\theta_j d\theta_j \\
&\quad + \pi(\ln(1-f_1))^2 + \pi(\ln f_1)^2]
\end{aligned}$$

By Lemma 3, $\int_0^{\frac{\pi}{2}} (\ln\cos\theta_j)^2 d\theta_j$ and $\int_0^{\frac{\pi}{2}} \ln\cos\theta_j d\theta_j$ are both finite. Thus,

$$\mathrm{E}[(\ln \psi_{S_1}(f, \boldsymbol{\theta_j}))^2] < \infty.$$

Therefore, $\mathrm{Var}(X_j)$ is finite. □

Now we present the asymptotic result on the probability that a split cut cuts off more volume than a type 1 triangle cut as the number of continuous variables increases.

Theorem 3.

$$\lim_{k\to\infty} \Pr[\mathrm{C}_{\mathrm{S}_1} \succ_{\mathrm{V}} \mathrm{C}_{\mathrm{T}_1}|\mathrm{f}] = \begin{cases} 1 & \textit{if } \mu_f < 0 \\ 1/2 & \textit{if } \mu_f = 0 \\ 0 & \textit{if } \mu_f > 0. \end{cases}$$

Proof. From Theorem 2, we know $\Pr[\mathrm{C}_{\mathrm{S}_1} \succ_{\mathrm{V}} \mathrm{C}_{\mathrm{T}_1}|\mathrm{f}] = \Pr[\sum_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0]$. Since $\boldsymbol{X_j}(f)$ $(j = 1, \ldots, k)$ are i.i.d., we can apply the Weak Law of Large Numbers and the Central Limit Theorem. Let $\overline{\boldsymbol{X}}_k(f) = \frac{\sum_{j=1}^{k} \boldsymbol{X_j}(f)}{k}$. Since $|\mu_f|$ is finite (Lemma 4), by the Weak Law of Large Numbers, $\lim\limits_{k\to\infty} \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| < \epsilon] = 1$ for any $\epsilon > 0$. We consider three cases: $\mu_f < 0$, $\mu_f > 0$ and $\mu_f = 0$.

(1) $\mu_f < 0$. Choose $\epsilon = -\dfrac{\mu_f}{2}$. Then

$$\Pr[\sum_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] = \Pr[\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) < 0] \geq \Pr[\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}} < \epsilon] \geq \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| < \epsilon].$$

Thus, $\liminf\limits_{k\to\infty} \Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] \geq \liminf\limits_{\mathrm{k}\to\infty} \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| < \epsilon] = \lim\limits_{\mathrm{k}\to\infty} \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| < \epsilon] = 1$. Since $\limsup\limits_{k\to\infty} \Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] \leq 1$, $\lim\limits_{k\to\infty} \Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] = 1$.

(2) $\mu_f > 0$. Choose $\epsilon = \dfrac{\mu_f}{2}$. Then

$$\Pr[\sum_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] = \Pr[\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) < 0] \leq \Pr[\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}} < -\epsilon] \leq \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| > \epsilon].$$

Thus, $\limsup\limits_{k\to\infty} \Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] \leq \limsup\limits_{\mathrm{k}\to\infty} \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| > \epsilon] = \lim\limits_{\mathrm{k}\to\infty} \Pr[|\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}| > \epsilon] = 0$. Since $\liminf\limits_{k\to\infty} \Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{X}_{\mathrm{j}} < 0] \geq 0$, $\Pr[\sum\limits_{\mathrm{j}=1}^{\mathrm{k}} \boldsymbol{X_j}(\mathrm{f}) < 0] = 0$.

(3) $\mu_f = 0$. From Lemma 4, σ_f^2 is finite. By the Central Limit Theorem, $\dfrac{\overline{\boldsymbol{X}}_k(f) - \mu_f}{\sigma_f/\sqrt{k}}$ converges to the standard normal random variable $\mathcal{N}(0,1)$ in distribution. Thus

$$\lim_{k\to\infty} \Pr[\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{X}_{\mathrm{j}} < 0] = \lim_{k\to\infty} \Pr[\frac{\overline{\boldsymbol{X}}_{\mathrm{k}}(\mathrm{f}) - \mu_{\mathrm{f}}}{\sigma_{\mathrm{f}}/\sqrt{\mathrm{k}}} < 0] = \frac{1}{2}.$$

□

Define $R_V(S_1, T_1) = \{f \in U : \mu_f < 0\}$ and $R_V(T_1, S_1) = \{f \in U : \mu_f > 0\}$. Then, $R_V(S_1, T_1)$ indicates the region where the split cut C_{S_1} cuts off more volume than the type 1 triangle cut C_{T_1} with probability close to 1 when k is large, and $R_V(T_1, S_1)$ indicates the region where the type 1 triangle cut C_{T_1} cuts off more volume than the split cut C_{S_1} with probability close to 1 when k is large. Even though $\boldsymbol{\theta_j}$ has a simple distribution, it is difficult to analytically compute μ_f. However we can estimate μ_f by Monte Carlo simulation for a given value of f, and identify the regions $R_V(S_1, T_1)$ and $R_V(T_1, S_1)$. The black and white regions in Figure 5 indicate $R_V(S_1, T_1)$ and $R_V(T_1, S_1)$, respectively. These have been identified as follows. First we randomly generate 10^5 fractional points f in U; then for each f, we independently generate 1000 θ_j uniformly from $[0, 2\pi)$ and check if the sample mean of $\ln \frac{\psi_{S_1}(f,\theta_j)}{\psi_{T_1}(f,\theta_j)}$ is less or greater than zero to identify if the corresponding f is in $R_V(S_1, T_1)$ or $R_V(T_1, S_1)$. The area of the black region is approximately 0.9. Unless f_1 is close to 1, the split cut C_{S_1} cuts off more volume than the type 1 triangle cut C_{T_1} with probability close to 1 when k is large, and therefore C_{S_1} is preferred.

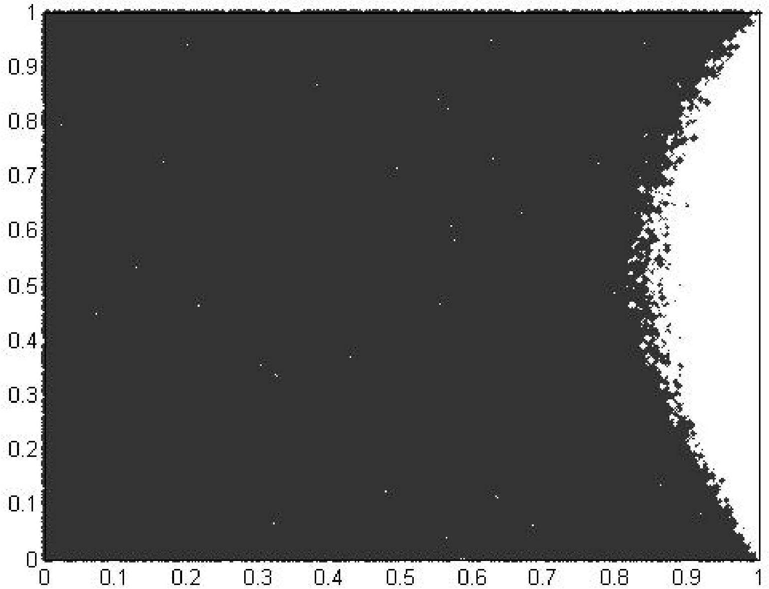


Figure 5: The shape of $R_V(S_1, T_1)$ and $R_V(T_1, S_1)$.

2.4 Total Probabilities

In this section, we use the conditional probabilities from the previous section to compute coefficient dominance and volume cut off probabilities for split and type 1 triangle cuts when f is random. As before, we focus on the split cut C_{S_1} and the type 1 triangle cut C_{T_1} and note that the analysis and conclusions extend to another pair of split and type 1 triangle bodies by symmetry. The total probability analysis provides some insight on how these cuts are likely to perform when no information about the instance is available.

2.4.1 Cut coefficient comparison

By the law of total probability,

$$\begin{aligned}\mathrm{Pr}[\mathrm{C_{S_1}} \succ_\mathrm{D} \mathrm{C_{T_1}}] &= \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j}),\ \forall \mathrm{j}] \\ &= \oint_U \mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j}),\ \forall \mathrm{j}|\mathrm{f}]\mathrm{d}\Phi(\mathrm{f}) \\ &= \oint_U \{\mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j})|\mathrm{f}]\}^\mathrm{k}\mathrm{d}\Phi(\mathrm{f}),\end{aligned}$$

where $\Phi(f)$ is the cumulative distribution function of $\mathbf{f}$ and the last equality follows from the fact that $\boldsymbol{\theta_j}$ are i.i.d. for $j = 1, \ldots, k$. Recall that the conditional probability $\mathrm{Pr}[\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j}) < \psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j})|\mathrm{f}]$ is given in Lemma 1. The following theorem describes the performance of the split cut C_{S_1} and type 1 triangle cut C_{T_1} when there is only one continuous variable.

Theorem 4. *If $k = 1$ then*

$$\mathrm{Pr}[\mathrm{C_{S_1}} \succ_\mathrm{D} \mathrm{C_{T_1}}] \approx 0.426 > 0.25 = \mathrm{Pr}[\mathrm{C_{T_1}} \succ_\mathrm{D} \mathrm{C_{S_1}}].$$

Proof. Note that $\angle BOC$, $\angle AOB$ and $\angle COF$ are shown in Figure 3. Then

$$\mathrm{Pr}[\mathrm{C_{T_1}} \succ_\mathrm{D} \mathrm{C_{S_1}}] = \oint_\mathrm{U} \mathrm{Pr}[\psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta}) < \psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta})]\mathrm{d}\Phi(\mathrm{f}) = \oint_\mathrm{U} \frac{\angle \mathrm{BOC}}{2\pi}\mathrm{d}\Phi(\mathrm{f}).$$

Similarly,

$$\mathrm{Pr}[\mathrm{C_{S_1}} \succ_\mathrm{D} \mathrm{C_{T_1}}] = \oint_\mathrm{U} \frac{\angle \mathrm{AOB} + \angle \mathrm{COF}}{2\pi}\mathrm{d}\Phi(\mathrm{f}).$$

The proof then follows from Lemma 5. □

Lemma 5.

$$\oint_U \frac{\angle BOC}{2\pi} d\Phi(f) = \oint_U \frac{\angle COD}{2\pi} d\Phi(f) = \oint_U \frac{\angle DOA}{2\pi} d\Phi(f) = \oint_U \frac{\angle AOB}{2\pi} d\Phi(f) = 0.25,$$

and

$$\oint_U \frac{\angle COF}{2\pi} d\Phi(f) \approx 0.176.$$

Proof. Indeed, since Φ is uniformly distributed over U,

$$\oint_U \frac{\angle COD}{2\pi} d\Phi(f) = \lim_{\epsilon \to 0} \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle COD}{2\pi} df_1 df_2$$

In $\triangle COD$, $|\overline{OC}| = \sqrt{(1-f_1)^2 + (1-f_2)^2}$, $|\overline{OD}| = \sqrt{f_1^2 + (1-f_2)^2}$ and $|\overline{CD}| = 1$. By the law of cosines,

$$\cos \angle COD = \frac{|\overline{OD}|^2 + |\overline{OC}|^2 - |\overline{CD}|^2}{2|\overline{OD}||\overline{OC}|} = \frac{f_1(f_1-1) + (1-f_2)^2}{\sqrt{[f_1^2 + (1-f_2)^2][(1-f_1)^2 + (1-f_2)^2]}}$$

Therefore, $\angle COD = \arccos \frac{f_1(f_1-1)+(1-f_2)^2}{\sqrt{[f_1^2+(1-f_2)^2][(1-f_1)^2+(1-f_2)^2]}}$. Similarly,

$$\angle BOC = \arccos \frac{f_2(f_2-1) + (1-f_1)^2}{\sqrt{[f_2^2 + (1-f_1)^2][(1-f_2)^2 + (1-f_1)^2]}}.$$

By substitution of variables,

$$\begin{aligned}
& \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle COD}{2\pi} df_1 df_2 \\
= & \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\arccos \frac{f_1(f_1-1)+(1-f_2)^2}{\sqrt{[f_1^2+(1-f_2)^2][(1-f_1)^2+(1-f_2)^2]}}}{2\pi} df_1 df_2 \\
\xrightarrow{f_1=1-g_2, f_2=g_1} = & \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\arccos \frac{g_2(g_2-1)+(1-g_1)^2}{\sqrt{[(1-g_2)^2+(1-g_1)^2][g_2^2+(1-g_1)^2]}}}{2\pi} dg_2 dg_1 \\
= & \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\arccos \frac{g_2(g_2-1)+(1-g_1)^2}{\sqrt{[g_2^2+(1-g_1)^2][(1-g_2)^2+(1-g_1)^2]}}}{2\pi} dg_1 dg_2 \\
\xrightarrow{g_1=f_1, g_2=f_2} = & \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\arccos \frac{f_2(f_2-1)+(1-f_1)^2}{\sqrt{[f_2^2+(1-f_1)^2][(1-f_2)^2+(1-f_1)^2]}}}{2\pi} df_1 df_2 \\
= & \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle BOC}{2\pi} df_1 df_2
\end{aligned}$$

Similarly, we can show

$$\int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle DOA}{2\pi} df_1 df_2 = \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle AOB}{2\pi} df_1 df_2 = \int_\epsilon^{1-\epsilon} \int_\epsilon^{1-\epsilon} \frac{\angle BOC}{2\pi} df_1 df_2$$

Therefore,

$$\begin{aligned}
&4\int_{\epsilon}^{1-\epsilon}\int_{\epsilon}^{1-\epsilon}\frac{\angle BOC}{2\pi}df_1df_2\\
&=\int_{\epsilon}^{1-\epsilon}\int_{\epsilon}^{1-\epsilon}\frac{\angle BOC+\angle COD+\angle DOA+\angle AOB}{2\pi}df_1df_2\\
&=\int_{\epsilon}^{1-\epsilon}\int_{\epsilon}^{1-\epsilon}\frac{2\pi}{2\pi}df_1df_2\\
&=(1-2\epsilon)^2
\end{aligned}$$

Thus, $\oint_U \frac{\angle BOC}{2\pi}d\Phi(f)=0.25$.

Now we compute $\oint_U \frac{\angle COF}{2\pi}d\Phi(f)$.

$$\begin{aligned}
\oint_U \frac{\angle COF}{2\pi}d\Phi(f) &= \lim_{\epsilon\to 0}\int_{\epsilon}^{1-\epsilon}\int_{\epsilon}^{1-\epsilon}\frac{\angle COF}{2\pi}df_1df_2\\
&=\lim_{\epsilon\to 0}\int_{\epsilon}^{1-\epsilon}\int_{\epsilon}^{1-\epsilon}\frac{\arccos\frac{f_1^2+f_2^2-f_1-3f_2+2}{\sqrt{[(1-f_2)^2+(1-f_1)^2][f_1^2+(2-f_2)^2]}}}{2\pi}df_1df_2\approx 0.176.
\end{aligned}$$

In the final step, we used the Matlab function 'dblquad' with $\epsilon=10^{-8}$ for the numerical calculation. □

Now we consider the case $k>1$.

Theorem 5. *For any* k, $\Pr[\mathrm{C}_{\mathrm{S}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{T}_1}]>\Pr[\mathrm{C}_{\mathrm{T}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{S}_1}]$.

Proof.

$$\begin{aligned}
\Pr[\mathrm{C}_{\mathrm{S}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{T}_1}] &= \oint_U\left(\frac{\angle AOB+\angle COF}{2\pi}\right)^k d\Phi(f)\\
&>\oint_U\left(\frac{\angle AOB}{2\pi}\right)^k d\Phi(f)=\oint_U\left(\frac{\angle BOC}{2\pi}\right)^k d\Phi(f)\\
&=\Pr[\mathrm{C}_{\mathrm{T}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{S}_1}].
\end{aligned}$$

The second equality follows from symmetry since f is uniformly distributed in $(0,1)^2$. □

Theorem 5 states that a single split cut is more likely to dominate a single type 1 triangle cut under our probabilistic model no matter how many continuous variables there are in system (1). We also use Monte Carlo simulation to estimate the magnitude of the probabilities that one cut dominates another. The result is shown in Figure 6.

From Figure 6, although $\Pr[\mathrm{C}_{\mathrm{S}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{T}_1}]>\Pr[\mathrm{C}_{\mathrm{T}_1}\succ_{\mathrm{D}}\mathrm{C}_{\mathrm{S}_1}]$ for all k, both probabilities are very small when $k\geq 5$ indicating that it is unlikely that one cut totally dominates another when there are many continuous variables.

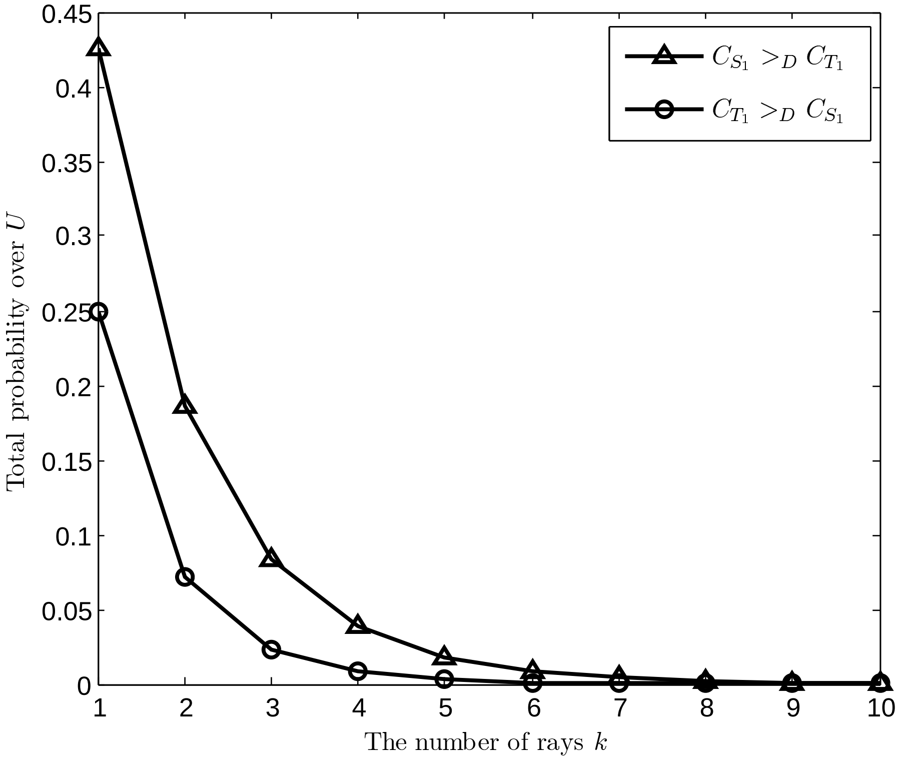


Figure 6: $\mathrm{Pr}[\mathrm{C_{S_1}} \succ_{\mathrm{D}} \mathrm{C_{T_1}}]$ and $\mathrm{Pr}[\mathrm{C_{T_1}} \succ_{\mathrm{D}} \mathrm{C_{S_1}}]$
wrt the number of rays k.

2.4.2 Volume comparison

In this section we estimate $\mathrm{Pr}[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}]$ with respect to the number of continuous variables k. Recall that $\Pr[C_{S_1} \succ_V C_{T_1}] = \Pr[\prod_{\mathrm{j=1}}^{\mathrm{k}} \frac{\psi_{\mathrm{S_1}}(\mathbf{f}, \boldsymbol{\theta_j})}{\psi_{\mathrm{T_1}}(\mathbf{f}, \boldsymbol{\theta_j})} < 1]$. We use Monte Carlo simulation to estimate the above probabilities as follows. For each $k \in \{1, \dots, 1000\}$, we randomly generate $N = 10^5$ samples of $f_1, f_2, \theta_1, \cdots, \theta_k$ according to our probabilistic input model. The probability $\Pr[\prod_{j=1}^{k} \frac{\psi_S(\mathbf{f}, \boldsymbol{\theta_j})}{\psi_{T_1}(\mathbf{f}, \boldsymbol{\theta_j})} < 1]$ is then estimated by the proportion of the N samples with $\prod_{j=1}^{k} \frac{\psi_S(\mathbf{f}, \boldsymbol{\theta_j})}{\psi_{T_1}(\mathbf{f}, \boldsymbol{\theta_j})} < 1$. The estimated probabilities with respect to k are shown in Figure 7. The estimated probability that C_{S_1} cuts off more volume from the linear relaxation than C_{T_1} increases as the number of continuous variables increases, converging to approximately 0.9. To explain this, note that

$$\lim_{k\to\infty} \mathrm{Pr}[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}] = \lim_{\mathrm{k}\to\infty} \oint_{\mathrm{U}} \mathrm{Pr}[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}|\mathrm{f}] \mathrm{d}\Phi(\mathrm{f}).$$

Since $\Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}|\mathrm{f}]$ is bounded, by interchanging limit and integral and applying Theorem 3 we have

$$\lim_{k\to\infty} \Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}] = \oint_{\mathrm{U}} \lim_{\mathrm{k}\to\infty} \Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}|\mathrm{f}]\mathrm{d}\Phi(\mathrm{f})$$

$$= \oint_U \{\mathbb{I}(\mu_f < 0) + \frac{1}{2}\mathbb{I}(\mu_f = 0)\}d\Phi(f) \geq \oint_U I(\mu_f < 0)d\Phi(f) = \Pr[\mathbf{f} \in \mathrm{R_V(S_1, T_1)}],$$

where $\mathbb{I}(A)$ is the indicator function of event A and $R_V(S_1, T_1)$ is defined in Section 2.3.2. Figure 7 presents $\Pr[\mathrm{C_{S_1}} \succ_{\mathrm{V}} \mathrm{C_{T_1}}]$ with respect to the number of continuous variables k (in two different scales). Recall that, as observed in Figure 5, the area of $R_V(S_1, T_1)$ is approximately 0.9, which coincides with the observation in Figure 7. We can conclude C_{S_1} is more likely to cut off more volume than C_{T_1} when k is not too small given any instance of (1) with parameters distributed according to our probabilistic input model.

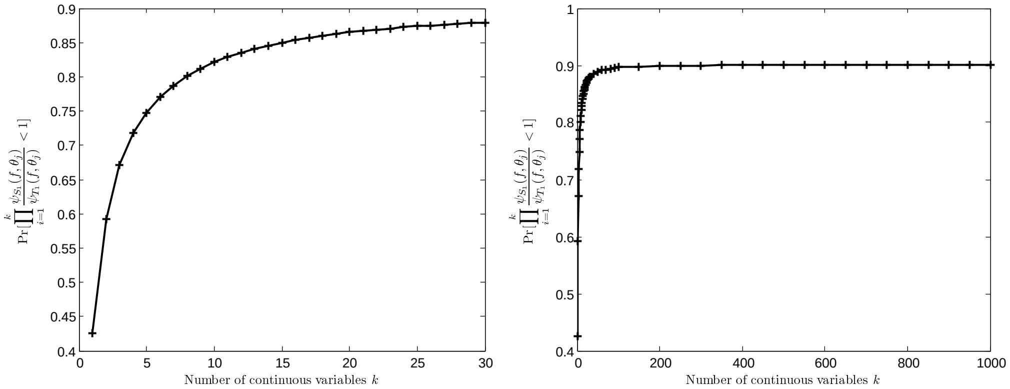


Figure 7: Estimated $\Pr[\prod_{\mathrm{j=1}}^{\mathrm{k}} \frac{\psi_{\mathrm{S}}(\mathbf{f},\boldsymbol{\theta_j})}{\psi_{\mathrm{T_1}}(\mathbf{f},\boldsymbol{\theta_j})} < 1]$ with respect to k.

2.5 Conclusions and future work

In this chapter, we proposed a probabilistic model to compare split cuts and type 1 triangle cuts. The analysis can be extended to other classes of facet-defining intersection cuts where the corresponding integer-free body contains the unit square, such as type 2 triangles and quadrilaterals containing U. In particular, for the comparison of volume cut off, similar results as in Theorem 2, Lemma 4 and Theorem 3 can be derived, since the type 2 triangles and quadrilaterals are all bounded and the corresponding cut coefficients are strictly greater than zero. Although it might be difficult to compute the associated probabilities analytically, we can still estimate the probability numerically and obtain regions of f where one cut dominates another or cuts off more volume. The analysis for type 3 triangles is much less obvious since such a triangle does not contain U. Another interesting question is how to extend our probabilistic analysis on cut comparisons to the model with explicit bounds on the y variables. In this model, the region cut off from the LP relaxation by an individual cut is not always a simplex, and therefore the volume comparison becomes more complicated. It would also be interesting to study how to extend our analysis on volume comparison to multiple rounds of cuts. Finally we note that, recently, two groups, Del Pia et al. [44] and Basu et al. [17], have also conducted probabilistic analyses of the strength of various families of two-row cuts, using different probabilistic models and comparison criteria.

CHAPTER III

MINIMUM CONCAVE COST FLOW OVER A GRID NETWORK

3.1 Introduction

We study the minimum concave cost flow problem over a grid network (CFG). A *grid network* $G = (V, A)$ is a directed acyclic graph with the node set

$$V = \{v_{l,t} | l \in \{0, \ldots, L\}, t \in \{1, \ldots, T\}\}$$

and the arc set

$$\begin{aligned} A = &\{(v_{l,t}, v_{l,t+1}) | l \in \{0, \ldots, L\}, t \in \{1, \ldots, T-1\}\} \cup \\ &\{(v_{l,t}, v_{l+1,t}) | l \in \{0, \ldots, L-1\}, t \in \{1, \ldots, T\}\} \end{aligned}$$

as show in Figure 8. The nodes and/or arcs have associated numerical values such as supplies/demands, costs and capacities. We refer to the two subscripts l and t as the indices of *echelon* and *time period*, respectively, so the grid network we study has $L + 1$ echelons and T time periods.

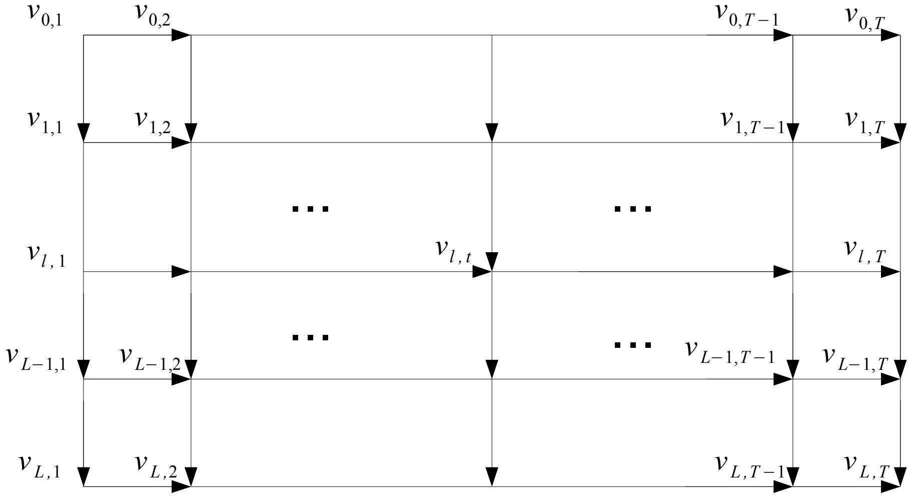


Figure 8: The grid network

Given a grid network G, CFG is to find a vector $x \in \mathbb{R}^{|A|}$ to

$$\begin{aligned} \min \quad & \textstyle\sum_{a\in A} c_a(x_a) & & \\ \text{s.t.} \quad & \textstyle\sum_{a\in\delta^+(v)} x_a - \sum_{a\in\delta^-(v)} x_a = b(v), & & \forall v \in V, \\ & x_a \geq 0, & & \forall a \in A, \end{aligned} \tag{8}$$

where c_a is the cost function for arc a, $b(v)$ is the supply at node v, and $\delta^+(v)$ and $\delta^-(v)$ are the set of outgoing and incoming arcs at node v, respectively. The node v is called a source if $b(v) > 0$, a sink if $b(v) < 0$, and a transshipment node if $b(v) = 0$. We assume that for each $a \in A$, the cost function c_a is a general nonnegative concave function represented by a value oracle.

The main contributions of this chapter are the following.

1. If all sources are at one echelon and all sinks are at L echelons with L fixed, then CFG can be solved in polynomial time in T and the number of queries of a function-value oracle,

2. If all sources are at one echelon and all sinks are at L echelons with L as an input parameter, then CFG is NP-hard.

3. The above complexity results can be extended to CFG with backward arcs in the grid network.

4. If there are upward arcs in the grid network, then CFG is NP-hard for any fixed $L \geq 2$.

Part of the results in this chapter (Section 3.2) has been submitted to *Mathematical Programming* [73].

The minimum concave cost network flow problem (MCCNFP) is NP-hard in general, as shown by a reduction from the subset sum problem [66], with a few known polynomially solvable special cases. There are two polynomially-solvable cases closely related to the problem we study. Zangwill [103] gave a polynomial-time DP algorithm for the multi-echelon lot-sizing problem, which can be formulated as a CFG with a single source and one echelon of sinks. Erickson et al. [53] proposed a DP algorithm for the general MCCNFP and showed that their algorithm runs in polynomial time when the graph is planar and all sources and sinks lie on a constant number of faces of the graph. The grid network is a planar graph, but in general the sources and sinks in CFG are not on a constant number of faces. Our result unifies the complexity results of the uncapacitated lot-sizing problem (ULS) and many of its variants including the multi-echelon case in Zangwill [103] and two-echelon lot-sizing problem with intermediate demands in Zhang et al. [104], and gives new complexity results for the multi-echelon ULS with arbitrary intermediate demand and multi-echelon pure remanufacturing problem with arbitrary intermediate demand. We provide two DP models for CFG with the component of the state vector being the inflow into some node and flow over some horizontal arc, respectively. We then derive a new characterization for the optimal flow in a grid network with multiple sources. The characterization leads to polytime DP algorithms, and also provides some insight on the optimal inventory and production quantities in a production planning setting. Our analysis was motivated by Zangwill's characterization of the optimal flow for the multi-echelon ULS. However, the presence of multiple sources introduces much more complexity on the structure of the optimal flow. Note that our result and the algorithm in Erickson et al. [53] both generalize the complexity result for multi-echelon ULS without intermediate demand, but we give new complexity results for CFG with arbitrary intermediate demand.

Apart from its theoretical interest, polynomial solvability of special cases of MCCNFP offers possibilities of deriving a tight or compact extended formulation for the original problem, which can help to solve the problem efficiently. For example, there is a general technique for deriving a compact extended formulation from a polytime DP algorithm in Martin [81, 80]. As a first step of the study, the main focus of this chapter is to discover general network topology over which the concave minimization problem can be efficiently solved, with few assumptions on the concave function itself. We are particularly interested in grid networks, which appear frequently as the underlying network structure or sub-structure in numerous business and engineering problems, such as the integrated supply chain management problem with coordination between manufacturer, distribution centers and retailers (Kaminsky and Simchi-Levi [75] and van Hoesel et al. [95]), production planning in a vertical production line (Pochet and Wolsey [84]), and remanufacturing and recycling problems in reverse logistics (Dekker et al. [43]), where the temporal dynamics of the system are modeled in one dimension of the grid network and the sequential actions in space are modeled in the other. The following three examples show how CFG generalizes other models.

The ULS model. The classical ULS is to determine an optimal production schedule given a sequence of deterministic non-stationary demand $d_1, \dots, d_T$ over T time periods with fixed setup cost α_t, unit production cost β_t and inventory holding cost i_t at period $t \in \{1, \dots, T\}$.

The ULS is a special case of CFG with $L = 1$, $b(v_{0,1}) = \sum_{t=1}^{T} d_t$, $b(v_{1,t}) = -d_t$ for $t \in \{1, \dots, T\}$, $b(v) = 0$ for any other node $v \in V$, and the cost function

$$c_a(x_a) = \begin{cases} \alpha_t \mathbb{I}(\{x_a > 0\}) + \beta_t x_a, & a = (v_{0,t}, v_{1,t}), t \in \{1, \dots, T\}, \\ i_t x_a, & a = (v_{1,t}, v_{1,t+1}), t \in \{1, \dots, T-1\}, \\ 0, & a = (v_{0,t}, v_{0,t+1}), t \in \{1, \dots, T-1\}, \end{cases} \tag{9}$$

where $\mathbb{I}(\{x_a > 0\})$ is an indicator function whose value is 1 if $x_a > 0$ and 0 otherwise. Variants of ULS, such as assuming the production and holding costs to be general concave functions, the presence of multiple echelons and intermediate demands, can be also formulated as CFG. ULS with backlogging can also be formulated as CFG by adding additional backward arcs into the grid network.

The pure remanufacturing model. Over T periods, a company recycles used products with quantities $p_1, \dots, p_T$ and remanufactures then into new products to satisfy the demands $d_1, \dots, d_T$. The pure remanufacturing problem is to find an optimal production schedule to minimize the total production and holding costs, given the remanufacturing cost function α_t, the holding cost functions $i_{0,t}$ and $i_{1,t}$ for used and new products, respectively. The model appears as a basic model in reverse logistics (Dekker et al. [43]). Although in van den Heuvel et al. [94] the model is shown to be equivalent to the ULS with inventory upper bounds when $i_{0,t}$ and $i_{1,t}$ are both linear functions, the transformation is not applicable when $i_{0,t}$ and $i_{1,t}$ are general concave functions. In fact, the model is more general since ULS can be seen as a special case of this model by letting $p_1 = \sum_{t=1}^{T} d_i$ and $p_t = 0$ for $2 \leq t \leq T$. The pure remanufacturing model is a special case of CFG with $L = 1$,

$b(v_{0,t}) = p_t$ and $b(v_{1,t}) = -d_t$ for $t \in \{1, \ldots, T\}$, and the cost function

$$c_a(x_a) = \begin{cases} \alpha_t(x_a), & a = (v_{0,t}, v_{1,t}), t \in \{1, \ldots, T\}, \\ i_{0,t}(x_a), & a = (v_{0,t}, v_{0,t+1}), t \in \{1, \ldots, T-1\}, \\ i_{1,t}(x_a), & a = (v_{1,t}, v_{1,t+1}), t \in \{1, \ldots, T-1\}. \end{cases} \tag{10}$$

To the best of our knowledge, the complexity of this problem was not known when there are multiple echelons with intermediate demands and general concave cost functions. In this chapter, we show that the problem is polynomially solvable when there are a constant number of echelons of intermediate demands, and NP-hard when the number of echelons of intermediate demands is an input parameter.

Production planning in a rolling horizon model. In a rolling horizon model for production planning where decisions are made periodically within a given time horizon, the presence of initial inventory is inevitable. However, the traditional setting for the lot-sizing problems assumes that the initial inventory is 0 at each echelon. The purpose of this critical assumption is more theoretical than practical, since it makes the underlying network a single-source network and greatly simplifies the analysis. When there are fixed initial inventory at each echelon in a multi-echelon setting, the analysis of the structure of the optimal solutions becomes cumbersome and difficult, as shown in van Hoesel et al. [95]. In fact, the rolling horizon model with initial inventory can be easily dealt with by transforming it to an equivalent CFG with $L-1$ additional time periods. Suppose that in the rolling horizon model the initial inventory at echelon l is I_l for $1 \leq l \leq L$. As shown in Figure 9, by attaching $L-1$ time periods before period 1, setting the supplies of the new sources at echelon 0 to be $I_L, I_{L-1}, \ldots, I_1$ and the cost to be large enough for each bold arc and 0 otherwise, the rolling horizon model with initial inventory is transformed into an equivalent CFG.

We end the introduction by reviewing MCCNFPs that can be solved in polynomial time. Most of them fall into the category of lot-sizing problems. The classical ULS model was first solved by Wagner and Whitin [99] in $O(T^2)$ time with DP, and the complexity was later improved to $O(T \ln T)$ in Aggarwal and Park [1], Federgruen and Tzur [55] and Wagelmans et al. [98]. Zangwill [103] gave a DP algorithm for the multi-echelon ULS with

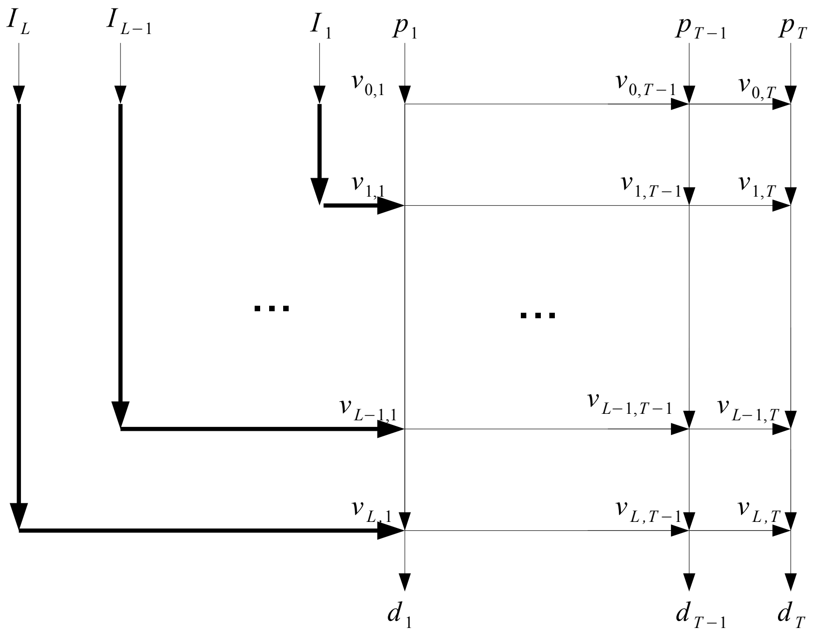


Figure 9: An equivalent CFG for the rolling horizon model with initial inventory

demands in the last echelon, and his algorithm was later shown to run in $O(LT^4)$ time for the L-echelon case in van den Heuvel et al. [95]. Other polynomial solvable variants include the constant capacitated lot-sizing problem (Florian and Klein [57]), ULS with backlogging, ULS with inventory upper bounds (Atamtürk and Küçükyavuz [6, 5]), a multi-echelon ULS with constant production capacities at the first echelon (van den Heuvel et al. [95]), and recently a two-echelon ULS with intermediate demands (Zhang et al. [104]). Pochet and Wolsey [84] provides a detailed study of lot-sizing models that can be solved in polytime. Besides the lot-sizing problem, polynomially solvable cases include a single-source concave network flow problem with a single nonlinear arc cost (Guisewite and Pardalos [67]), the network flow problem with a fixed number of sources and nonlinear arc costs (Tuy et al. [92]), and a production-transportation network flow problem where the concave cost function is defined on only a constant number of variables (Tuy et al. [93]).

3.2 CFG with at most two echelons of sinks.

In this section, we provide a DP framework for CFG with the component of the state variable being the inflow into each node, and show that CFG with at most two echelons of sinks can be solved in polynomial time when the total number of echelons is fixed.

3.2.1 The DP framework

We assume that the number of echelons L is a constant and $L \ll T$, which is reasonable in practice since L is usually known in advance and we are more interested in the complexity of CFG with respect to the number of time periods T. We propose to solve CFG by using a discrete time DP. The difficulty is that the state space at each stage of a natural DP formulation is an uncountable set. However, by analyzing the structure of the extreme points of the feasible set of (8), we are able to refine the state space to a set of size polynomial in T. For ease of exposition, we call the arcs of the form $(v_{l,t}, v_{l,t+1})$ the *forward* arcs and the arcs of the form $(v_{l,t}, v_{l+1,t})$ the *downward* arcs. To be consistent with the index of echelon in the lot-sizing model, we assume in our model that the first echelon is echelon 0. The elements of the DP are as follows.

1. Decision stages. There are $L + T$ stages, and the nodes with $l + t = k$ are at stage k, as shown in Figure 10.

2. States. Define the state $\mathbf{s}^k$ at stage k to be a vector whose component s_i^k denotes the amounts of inflow into some node $v_{l,t}$ with $l + t = k$. For example, $\mathbf{s}^1$ is a scalar that denotes the inflow into node $v_{0,1}$ and $\mathbf{s}^2$ is a two-dimensional vector whose first component is the inflow into node $v_{1,1}$ and second component is the inflow into node $v_{0,2}$. In general, when $k \leq L + 1$, $\mathbf{s}^k$ is a k-dimensional vector whose components are the amounts of inflow into nodes $v_{k-1,1}, v_{k-2,2}, \ldots, v_{0,k}$; when $L + 1 \leq k \leq T$, $\mathbf{s}^k$ is a $(L + 1)$-dimensional vector whose components are the amount of inflow into nodes $v_{L,k-L}, v_{L-1,k-L+1}, \ldots, v_{0,k}$; when $k \geq T+1$, $\mathbf{s}^k$ is a $(L+T+1-k)$-dimensional vector whose components are the amounts of inflow into nodes $v_{L,k-L}, \ldots, v_{k-T,T}$.

3. Decision variables (or actions). Let the decision variable $\mathbf{u}^k$ at stage k be a vector whose components are the amount of flow sent out by nodes at stage k through *downward* arcs. For example, $\mathbf{u}^1$ is a scalar which denotes the flow sent along the arc $(v_{0,1}, v_{1,1})$ and $\mathbf{u}^2$ is a two-dimensional vector whose first component is the flow on arc $(v_{1,1}, v_{2,1})$ and second component is the flow on arc $(v_{0,2}, v_{1,2})$. In general, when $k \leq L + 1$, $\mathbf{u}^k$ is a k-dimensional vector whose components are flows on arcs

$(v_{k-1,1}, v_{k,1})$, $(v_{k-2,2}, v_{k-1,2})$,. . .,$(v_{0,k}, v_{1,k})$; when $L+1 \le k \le T$, $\mathbf{u}^k$ is a L-dimensional vector whose components are flows on arcs $(v_{L-1,k+1-L}, v_{L,k+1-L})$, . . ., $(v_{0,k}, v_{1,k})$; when $T+1 \le k \le L+T-1$, $\mathbf{u}^k$ is a $(L+T-k)$-dimensional vector whose components are flows on arcs $(v_{L-1,k+1-L}, v_{L,k+1-L})$, . . ., $(v_{k-T,T}, v_{k+1-T,T})$.

4. The system equations. The state $\mathbf{s}^{k+1}$ at stage $k+1$ can be easily calculated by the flow balance constraints once the state $\mathbf{s}^k$ and the decision variable $\mathbf{u}^k$ are known. Let the system equations be $\mathbf{s}^{k+1} = H_k(\mathbf{s}^k, \mathbf{u}^k)$, where H_k is the affine function representing the flow balance constraints for nodes at stage $k+1$.

5. The cost function. The cost at stage k is the sum of all costs incurred by the downward arcs and forward arcs connecting nodes at stage k and nodes at stage $k+1$, so it is a function of $\mathbf{u}^k$ and $\mathbf{s}^k$. We use the function $r_k(\mathbf{s}^k, \mathbf{u}^k)$ to denote the cost incurred at stage k.

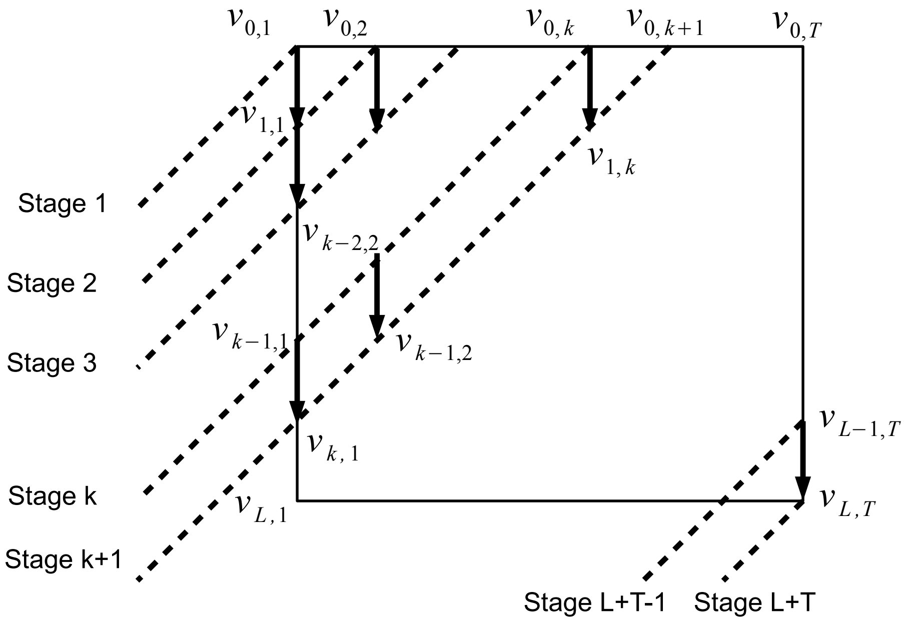


Figure 10: The dynamic programming formulation of CFG

Then CFG is formulated as a discrete time DP problem with the linear system $\mathbf{s}^{k+1} = H_k(\mathbf{s}^k, \mathbf{u}^k)$ and cost function r_k over $L+T$ stages. This DP formulation is difficult to solve directly since the state space at stage k is an uncountable set in general. However, by (8)

CFG is equivalent to minimizing a concave function over the flow polyhedron $P_F := \{x \in \mathbb{R}^{|A|} | x \text{ satisfies constraints in (8)}\}$. It is well known that there exists an optimal solution which is an extreme point of P_F if P_F is not empty. Therefore in the DP formulation, it suffices to consider those states corresponding to the extreme points of P_F, the number of which is finite. To argue that this DP formulation can be solved in polynomial time, it remains to show that the cardinality of the state space at each stage is polynomial in T. Since the dimension of the state vector at each stage is at most $L + 1$, the task is reduced to show that each component of the state vector, namely the inflow into each node under all extreme points of P_F, can take on a finite set of values whose cardinality is polynomial in T.

Before proceeding to characterize the inflow under all extreme points, we introduce some terminology and notation which will be used throughout the chapter. Let $G = (V, A)$ be a directed graph or digraph. A *path* in G is an alternating sequence of distinct nodes and arcs $\{v_1, a_1, v_2, a_2, \ldots, v_l\}$ with $a_i = (v_{i-1}, v_i)$ for $1 \leq i \leq l - 1$. A *cycle* is a path $\{v_1, a_1, v_2, a_2, \ldots, v_l\}$ together with the arc (v_l, v_1). The concepts of path and cycle in an undirected graph are similar to their directed versions except without specifying arc directions. We will use the same term "path" or "cycle" to refer to the object in a directed or undirected graph when the context is clear. The induced subgraph of G by the arc set $A' \subseteq A$ is the subgraph $G' = (V', A')$ where V' consists of nodes incident to any arc in A'. A vector $\mathbf{f} \in \mathbb{R}^{|A|}$ is called a *flow* in G if $\mathbf{f}$ satisfies the constraints in (8). A flow $\mathbf{f}$ is called an *extreme flow* if it is an extreme point of the underlying flow polyhedron P_F. For any flow $\mathbf{f}$, let $A_f = \{a \in A | f_a > 0\}$ be the set of arcs with nonzero flow. Let G_f denote the subgraph of G induced by the arc set A_f. The *underlying undirected graph* of G_f is an undirected graph obtained by replacing all directed arcs of G_f with undirected edges.

We begin to characterize the extreme flows with the following proposition.

Proposition 1. *Each extreme flow* $\mathbf{f}$ *in* G *can be decomposed into flows along paths each of which starts at one source and ends at one sink. In such a decomposition, there is at most one path with positive flow between each source-sink pair.*

Proof. Every flow can be decomposed into flows along paths and cycles, where each path starts from a source and ends at a sink; see Ahuja et al. [2]. When $\mathbf{f}$ is an extreme flow, the underlying undirected graph of G_f does not contain any cycles, so in any flow decomposition there will be no cycle with positive flow and at most one path with positive flow between each source-sink pair. □

Proposition 1 provides an alternative way to calculate the inflow into a node under an extreme flow $\mathbf{f}$ rather than summing up the flows over incoming arcs of that node: first decompose $\mathbf{f}$ into flows along paths between source-sink pairs, and then the inflow into a node is the summation of flows along paths that contain that node under that decomposition. For example, part (a) of Figure 11 is a CFG with $L = 2$ and $T = 3$ and part (b) is one extreme flow. To calculate the inflow into the central node (which is 9), we can use either the flow decomposition in part (c) to obtain $9 = 3 + 3 + 3$, or the flow decomposition in part (d) to obtain $9 = 3 + 6$.

Note that the flow decomposition is not unique for the extreme flow $\mathbf{f}$. Our remaining job is to choose a *particular* flow decomposition under which it is easy to argue that the inflow can only attain a polynomial number of values.

3.2.2 CFG with sources at echelon 0 and sinks at echelon L.

We illustrate the idea of choosing the particular flow decomposition by a special case of CFG, where all sources are at echelon 0, all sinks are at echelon L and all other nodes are transshipment nodes. We call this case CFG-1. As mentioned earlier, CFG-1 generalizes the multi-echelon ULS, and is a special case of planar graphs with sources and sinks lying in a fixed number of faces studied in Erickson et al. [53]. For CFG-1, the inflow into any node under all extreme flows has a nice closed-form formula which generalizes the result for the multi-echelon ULS in Zangwill [103].

We introduce some notation first. Let the supply at node $v_{0,t}$ be p_t and the demand at node $v_{L,t}$ be d_t for every $t \in \{1, \ldots, T\}$. Let $P_t = \sum_{i=1}^{t} p_i$ and $D_t = \sum_{i=1}^{t} d_i$ be the cumulative supply and demand up to period $t \in \{1, \ldots, T\}$, respectively. Without loss of generality, we can always assume that $P_T = D_T$. Let $\Gamma = \cup_{t=1}^{T}\{D_t, P_t\} \cup \{0\}$. Then Γ

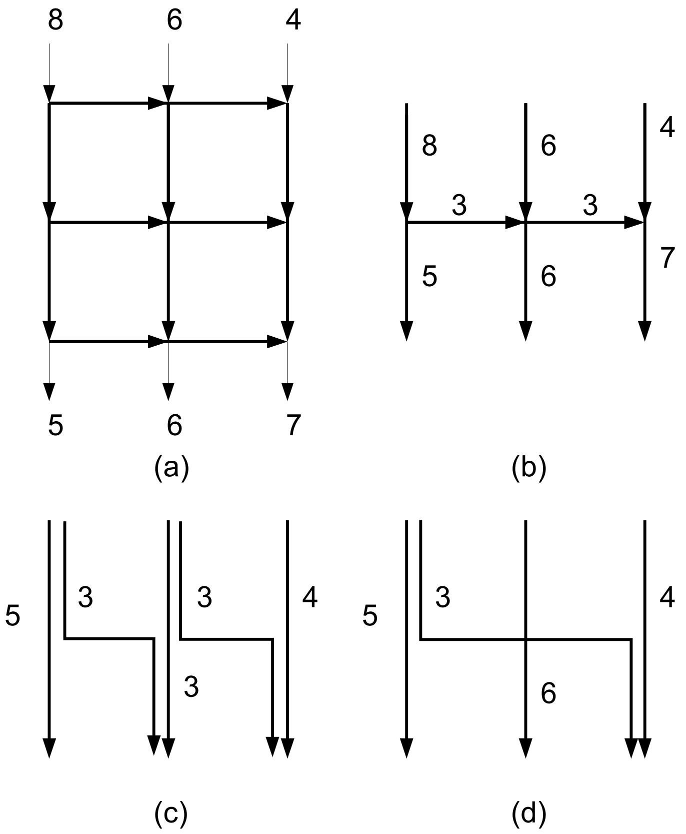


Figure 11: An example showing two flow decompositions for the same extreme flow.

contains at most $2T$ elements. The main technical result of this section is given below and we prove it later in the section.

Theorem 6. *For CFG-1, the inflow into any node in G under all extreme flows is $\gamma_2 - \gamma_1$, where $\gamma_1, \gamma_2 \in \Gamma$ with $\gamma_2 \geq \gamma_1$.*

Remark 1. Consider the multi-echelon ULS where the supply $p_1 = \sum_{t=1}^{T} d_t$, $p_2 = \ldots = p_T = 0$. The set $\Gamma = \{D_1, D_2, \ldots, D_T\}$. By applying Theorem 6, the possible values for the inflow into any node under all extreme flows are $D_{t_2} - D_{t_1} = \sum_{i=t_1+1}^{t_2} d_i$ with $t_1 \leq t_2$, a key result derived by Zangwill [103] in designing his DP algorithm for the multi-echelon ULS.

Theorem 7. *For fixed L, CFG-1 can be solved in polynomial time in T and the number of queries of a function-value oracle.*

Proof. By Theorem 6, the inflow into any node can attain $O(T^2)$ values under all extreme flows in CFG-1. Then in the DP formulation for CFG-1, the cardinality of the state space at each stage is $O(T^{2L})$, so there are $O(T^{4L})$ available actions at each stage. Since the DP has $L + T$ stages, CFG-1 can be formulated as a shortest path problem over an acyclic graph with $O(T^{4L+1})$ arcs, which can be solved in $O(T^{4L+1})$ time; see Ahuja et al. [2]. □

In the remainder of this section, we prove Theorem 6. As mentioned before, the idea is to choose a particular flow decomposition under which the inflow calculation is simple. By Proposition 1, there is at most one path with positive flow between any source-sink pair under any flow decomposition for a give extreme flow $\mathbf{f}$. Define $\lambda_f(i, j)$ to be the amount of flow sent along the path between the source $v_{0,i}$ and the sink $v_{L,j}$ under some decomposition for $\mathbf{f}$ (set $\lambda_f(i, j) = 0$ if there is no path from $v_{0,i}$ to $v_{L,j}$ in G_f), and consider the vector

$$\lambda_f = (\lambda_f(1, 1), \ldots, \lambda_f(1, T), \lambda_f(2, 1), \ldots, \lambda_f(2, T), \ldots, \lambda_f(T, 1), \ldots, \lambda_f(T, T)).$$

Then each flow decomposition for $\mathbf{f}$ can be represented by a vector λ_f. For example, the flow decompositions in part (c) and (d) of Figure 11 can be represented by $(5, 3, 0, 0, 3, 3, 0, 0, 4)$ and $(5, 0, 3, 0, 6, 0, 0, 0, 4)$, respectively. In the inflow calculation, we will choose the flow decomposition whose representation vector is the *lexicographically largest* among all flow

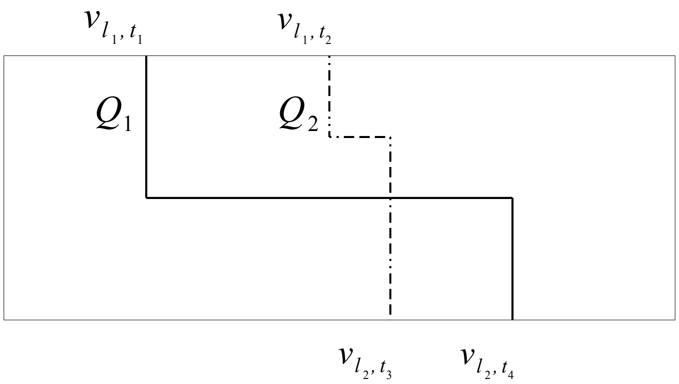


Figure 12: The two paths $\mathcal{Q}_1$ and $\mathcal{Q}_2$ must intersect.

decomposition vectors. Such a vector must exist since the set of all flow decomposition vectors is closed and bounded from above. We first give a formal definition of the lexicographical order between two vectors.

Definition 3. Given two vectors $\mu, \nu \in \mathbb{R}^n$, μ is lexicographically larger than ν, denoted by $\mu \succ \nu$, if there exists $i \in \{1, \ldots, n\}$ such that $\mu_j = \nu_j$ for $j \leq i-1$ and $\mu_i > \nu_i$; μ is lexicographically no smaller than ν, denoted by $\mu \succeq \nu$, if $\mu \succ \nu$ or $\mu = \nu$; μ is lexicographically smaller than (no larger than) ν, denoted by $\mu \prec (\preceq)\nu$, if $-\mu \succ (\succeq) - \nu$.

Let π_f be the lexicographically largest vector among all flow decomposition vectors for $\mathbf{f}$. For the example in Figure 11, the representation vector of the flow decomposition in part (c) is lexicographically largest for the extreme flow in part (b). We make the following simple observation, which will be useful to prove some nice properties of π_f.

Observation 1. *As illustrated in Figure 12, given a grid network G and four nodes v_{l_1,t_1}, v_{l_1,t_2}, v_{l_2,t_3} and v_{l_2,t_4} in G with $l_1 < l_2$, $t_1 \leq t_2$ and $t_3 \leq t_4$, let $\mathcal{Q}_1$ be any path from v_{l_1,t_1} to v_{l_2,t_4} and $\mathcal{Q}_2$ be any path from v_{l_1,t_2} to v_{l_2,t_3}. Then $\mathcal{Q}_1$ and $\mathcal{Q}_2$ must intersect.*

Now we begin to characterize the flow decomposition π_f.

Proposition 2.

1. *For any $i_1 < i_2$ and $j_1 < j_2$, $\pi_f(i_1, j_2) \cdot \pi_f(i_2, j_1) = 0$.*

2. *If $\pi_f(i_1, j_1) > 0$ and $\pi_f(i_1, j_2) > 0$ with $j_1 < j_2 - 1$, then $\pi_f(i_1, j) = d_j$ for any $j \in \{j_1 + 1, \ldots, j_2 - 1\}$.*

3. *If $\pi_f(i_1, j_1) > 0$ and $\pi_f(i_2, j_1) > 0$ with $i_1 < i_2 - 1$, then $\pi_f(i, j_1) = p_i$ for any $i \in \{i_1 + 1, \dots, i_2 - 1\}$.*

Proof. 1. Suppose that there exist $i_1 < i_2$ and $j_1 < j_2$ such that $\pi_f(i_1, j_2) \cdot \pi_f(i_2, j_1) > 0$. By Observation 1, the path from v_{0,i_1} to v_{L,j_2} must intersect with the path from v_{0,i_2} to v_{L,j_1}. If $\pi_f(i_1, j_2) \geq \pi_f(i_2, j_1)$, create a vector $\tilde{\pi}_f$ in the following way: $\tilde{\pi}_f(i_1, j_1) = \pi_f(i_1, j_1) + \pi_f(i_2, j_1)$, $\tilde{\pi}_f(i_1, j_2) = \pi_f(i_1, j_2) - \pi_f(i_2, j_1)$, $\tilde{\pi}_f(i_2, j_1) = 0$, $\tilde{\pi}_f(i_2, j_2) = \pi_f(i_2, j_2) + \pi_f(i_2, j_1)$ and $\tilde{\pi}_f(i, j) = \pi_f(i, j)$ for other (i, j) pairs. The vector $\tilde{\pi}_f$ represents another flow decomposition of $\mathbf{f}$ with $\tilde{\pi}_f \succ \pi_f$, a contradiction to the fact that π_f is the lexicographically largest flow decomposition vector. Similarly there is a contradiction when $\pi_f(i_2, j_1) \geq \pi_f(i_1, j_2)$.

2. Since $\pi_f(i_1, j_1) > 0$, by statement 1 $\pi_f(i, j) = 0$ for any $i < i_1$ and $j > j_1$. Since $\pi_f(i_1, j_2) > 0$, by statement 1 $\pi_f(i, j) = 0$ for any $i > i_1$ and $j < j_2$. Thus $\pi_f(i, j) = 0$ for each $j \in \{j_1 + 1, \dots, j_2 - 1\}$ and $i \neq i_1$. Then $\pi_f(i_1, j) = d_j$ for any $j \in \{j_1 + 1, \dots, j_2 - 1\}$ by the flow balance constraints.

3. Follows from a similar argument as in the proof of statement 2.

□

Proposition 2 shows that under this particular flow decomposition π_f, supply at each period is decomposed to satisfy demand from consecutive periods (statement 2), demand at each period is decomposed to be fulfilled by supply from consecutive periods (statement 3), and demand at an early period is always served as much as possible by supply at an early period (follows from statement 1).

In fact, the value of π_f can be computed exactly. As shown in Figure 13, put all the cumulative demand and supply points on the real line. Let $E_{i,j} = [P_{i-1}, P_i] \cap [D_{j-1}, D_j]$ for $i, j \in \{1, \dots, T\}$ and $\Delta_{i,j} = |E_{i,j}|$ denote the length of the interval $E_{i,j}$.

Proposition 3. *The vector π_f is fixed for any extreme flow $\mathbf{f}$, and $\pi_f(i, j) = \Delta_{i,j}$ for $1 \leq i, j \leq T$.*

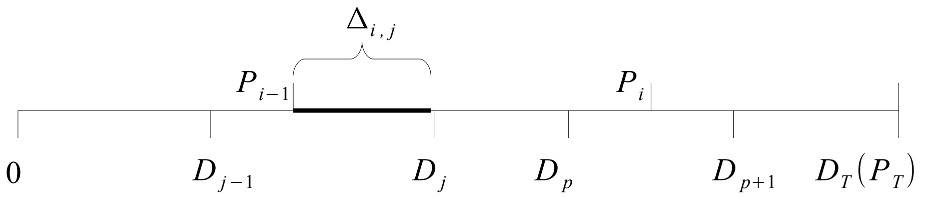


Figure 13: The supply-demand partition.

Proof. The proof is based on induction on the pair (i, j). For the base case where $(i, j) = (1, 1)$, $\pi_f(1, 1) = d_1$ since the sink $v_{L,1}$ can only be reached from the source $v_{0,1}$. Meanwhile, $P_1 \geq D_1$ if CFG-1 is feasible. Then $\Delta_{1,1} = d_1$ by the definition of $\Delta_{1,1}$. We have $\pi_f(1, 1) = \Delta_{1,1}$.

Now suppose that $\pi_f(i, j) = \Delta_{i,j}$ holds for all pairs $(i, j) \preceq (i_1, j_1)$. If $j_1 = T$, the next lexicographically larger pair is $(i_1 + 1, 1)$. Since there is no path from v_{0,i_1+1} to $v_{L,1}$, $\pi_f(i_1 + 1, 1) = 0$. Since $P_{i_1} \geq P_1 \geq D_1$, $\Delta_{i_1+1,1} = |[P_{i_1}, P_{i_1+1}] \cap [0, D_1]| = 0$. Then $\pi_f(i_1 + 1, 1) = \Delta_{i_1+1,1}$. If $j_1 < T$, the next lexicographically larger pair is $(i_1, j_1 + 1)$. We will show that $\pi_f(i_1, j_1 + 1) = \Delta_{i_1,j_1+1}$ in four different cases. WLOG, we assume that the supply $p_t > 0$ and demand $d_t > 0$ for $1 \leq t \leq T$. Recall that $\Delta_{i,j} = 0$ implies that either $P_{i-1} \geq D_j$ or $D_{j-1} \geq P_i$ and $\Delta_{i,j} > 0$ implies that $P_{i-1} < D_j$ or $D_{j-1} < P_i$.

1. If $\Delta_{i_1,j_1} > 0$ and $\Delta_{i_1,j_1+1} = 0$, by the definition of $\Delta_{i,j}$ we have $D_{j_1-1} < P_{i_1} \leq D_{j_1}$. Then

$$p_{i_1} = |[P_{i_1-1}, P_{i_1}]| = |[P_{i_1-1}, P_{i_1}] \cap \cup_{j=1}^{j_1} [D_{j-1}, D_j]| = \sum_{j=1}^{j_1} \Delta_{i_1,j} = \sum_{j=1}^{j_1} \pi_f(i_1, j).$$

The third equality follows from the definition of $\Delta_{i,j}$ and the last equality follows from the induction hypothesis. By the flow balance constraint at the node v_{0,i_1}, we have $\pi_f(i_1, j_1 + 1) = 0$. Then $\pi_f(i_1, j_1 + 1) = \Delta_{i_1,j_1+1}$.

2. If $\Delta_{i_1,j_1} > 0$ and $\Delta_{i_1,j_1+1} > 0$, by the definition of $\Delta_{i,j}$ we have $P_{i_1-1} < D_{j_1} < P_{i_1}$.

   (a) If $P_{i_1} > D_{j_1+1}$, then $\Delta_{i_1,j_1+1} = d_{j_1+1}$. Meanwhile,

$$\sum_{j=j_1+1}^{T} \pi_f(i_1, j) = p_{i_1} - \sum_{j=1}^{j_1} \pi_f(i_1, j) = p_{i_1} - \sum_{j=1}^{j_1} \Delta_{i_1,j} > \Delta_{i_1,j_1+1} = d_{j_1+1}. \quad (11)$$

The first equality is the flow balance constraint at the node v_{0,i_1} and the inequality follows from the assumption that $P_{i_1} > D_{j_1+1}$. Since $\pi_f(i_1, j_1+1) \leq d_{j_1+1}$, by (11) there must exists $j > j_1+1$ such that $\pi_f(i_1, j) > 0$. By applying statement 2 in Proposition 2 with $\pi_f(i_1, j_1) > 0$, we have $\pi_f(i_1, j_1+1) = d_{j_1+1}$. Then $\pi_f(i_1, j_1+1) = \Delta_{i_1, j_1+1}$.

(b) If $P_{i_1} \leq D_{j_1+1}$, $p_{i_1} = \sum_{j=1}^{j_1+1} \Delta_{i_1,j}$. Then $\sum_{j=j_1+1}^{T} \pi_f(i_1, j) = p_{i_1} - \sum_{j=1}^{j_1} \pi_f(i_1, j) = \Delta_{i_1,j_1+1}$. If $\pi_f(i_1, j_1+1) < \Delta_{i_1,j_1+1}$, then $\exists j > j_1+1$ such that $\pi_f(i_1, j) > 0$. By applying statement 2 in Proposition 2 with $\pi_f(i_1, j_1) > 0$, $\pi_f(i_1, j_1+1) = d_{j_1+1}$. Then $\pi_f(i_1, j_1+1) \geq \Delta_{i_1,j_1+1}$, a contradiction.

3. If $\Delta_{i_1,j_1} = 0$ and $\Delta_{i_1,j_1+1} = 0$, either $P_{i_1-1} \geq D_{j_1+1}$ or $D_{j_1-1} \geq P_{i_1}$. If $P_{i_1-1} \geq D_{j_1+1}$, $d_{j_1+1} = \sum_{i=1}^{i_1-1} \Delta_{i,j_1+1} = \sum_{i=1}^{i_1-1} \pi_f(i, j_1+1)$. Then $\pi_f(i_1, j_1+1) = 0$ by the flow balance constraint at node v_{L,j_1+1}. If $D_{j_1-1} \geq P_{i_1}$, then $p_{i_1} = \sum_{j=1}^{j_1-1} \Delta_{i_1,j} = \sum_{j=1}^{j_1-1} \pi_f(i_1, j)$. Then $\pi_f(i_1, j_1+1) = 0$ by the flow conservation constraint at node v_{0,i_1}.

4. If $\Delta_{i_1,j_1} = 0$ and $\Delta_{i_1,j_1+1} > 0$, then $D_{j_1} \leq P_{i_1-1} < D_{j_1+1}$.

(a) If $P_{i_1} > D_{j_1+1}$, $d_{j_1+1} = \sum_{i=1}^{i_1} \Delta_{i,j_1+1} = \sum_{i=1}^{i_1-1} \pi_f(i, j_1+1) + \Delta_{i_1,j_1+1}$. Since $d_{j_1+1} = \sum_{i=1}^{T} \pi_f(i, j_1+1)$, $\sum_{i=i_1}^{T} \pi_f(i, j_1+1) = \Delta_{i_1,j_1+1}$. Suppose that $\exists i > i_1$ such that $\pi_f(i, j_1+1) > 0$. Since $P_{i_1} > D_{j_1+1}$, $\exists j > j_1+1$ such that $\pi_f(i_1, j) > 0$. We find $i > i_1$ and $j > j_1+1$ such that $\pi_f(i, j_1+1) \cdot \pi_f(i_1, j) > 0$, a contradiction to statement 1 in Proposition 2. Then $\pi_f(i, j_1+1) = 0$ for all $i > i_1$ and $\pi_f(i_1, j_1+1) = \Delta_{i_1,j_1+1}$.

(b) If $P_{i_1} \leq D_{j_1+1}$, we have $D_{j_1} \leq P_{i_1-1} \leq P_{i_1} \leq D_{j_1+1}$. Then $\Delta_{i_1,j_1+1} = p_{i_1}$ and $\Delta_{i_1,j} = 0$ for each $j \leq j_1$, so $\pi_f(i_1, j) = 0$ for any $j \leq j_1$ by the induction hypothesis. Then $p_{i_1} = \sum_{j=j_1+1}^{T} \pi_f(i_1, j)$. If $\pi_f(i_1, j_1+1) < \Delta_{i_1,j_1+1} = p_{i_1}$, then $\exists j > j_1+1$ such that $\pi_f(i_1, j) > 0$ by the flow conservation constraint at node v_{0,i_1} and $\exists i > i_1$ such that $\pi_f(i, j_1+1) > 0$ by the flow conservation constraint at node v_{L,j_1+1}. We find $i > i_1$ and $j > j_1+1$ such that $\pi_f(i, j_1+1) \cdot \pi_f(i_1, j) > 0$, a contradiction to statement 1 in Proposition 2. Then $\pi_f(i_1, j_1+1) = \Delta_{i_1,j_1+1}$.

□

Proposition 3 is somewhat surprising. It states that the amount of flow sent between each source-sink pair is a constant under this particular flow decomposition, no matter what the flow $\mathbf{f}$ is. Thus we have an invariant quantity among all extreme flows, which is key to showing that the inflow can only attain a polynomial number of values in T. We present one more result before proving Theorem 6.

Proposition 4. *Given any extreme flow* $\mathbf{f}$*, let* $\mathcal{Q}_1$ *be a path from* v_{l_1,t_1} *to* v_{l_2,t_3} *and* $\mathcal{Q}_2$ *be a path from* v_{l_1,t_2} *to* v_{l_2,t_4} *in* G_f *with* $l_1 < l_2, t_1 \leq t_2$ *and* $t_3 \leq t_4$*. If* $\mathcal{Q}_1$ *and* $\mathcal{Q}_2$ *both contain* $v_{l,t}$*, then any path from* $v_{l_1,i}$ *to* $v_{l_2,j}$ *in* G_f *with* $t_1 \leq i \leq t_2$ *and* $t_3 \leq j \leq t_4$ *also contains the node* $v_{l,t}$*.*

Proof. Proof by contradiction. As shown in Figure 14, the node $v_{l,t}$ is contained in the path from v_{l_1,t_1} to v_{l_2,t_3} and the path from v_{l_1,t_2} to v_{l_2,t_4} in G_f. Suppose that there exists some pair (i,j) with $t_1 \leq i \leq t_2$ and $t_3 \leq j \leq t_4$ such that the path from $v_{l_1,i}$ to $v_{l_2,j}$ bypasses the node $v_{l,t}$. Then the path must contain some node $v_{l,u}$ with either $u < t$ or $u > t$. If $u < t$, by Observation 1, the path from v_{l_1,t_1} to $v_{l,t}$ must intersect with the path from $v_{l_1,i}$ to $v_{l,u}$, and the path from $v_{l,t}$ to v_{l_2,t_3} must intersect with the path $v_{l,u}$ to $v_{l_2,j}$ in G_f. The two intersections create a cycle in the underlying undirected graph of G_f, a contradiction. The argument is essentially the same if $u > t$.

□

Proof of Theorem 6. Given an extreme flow $\mathbf{f}$, the inflow to node $v_{l,t}$ can be calculated under the flow decomposition π_f as a summation of flows along paths that contain $v_{l,t}$. Let (i_1, j_1) and (i_2, j_2) be the lexicographically smallest and largest pairs (i,j) such that $\pi_f(i,j) > 0$ and the path from $v_{0,i}$ to $v_{L,j}$ in G_f contain $v_{l,t}$. Since $i_1 \leq i_2$ and $\pi_f(i_1,j_1), \pi_f(i_2,j_2) > 0$, by statement 1 of Proposition 2 we have $j_1 \leq j_2$. By applying Proposition 4 with $l_1 = 0$ and $l_2 = L$, any path from $v_{0,i}$ to $v_{L,j}$ in G_f with $i_1 \leq i \leq i_2$ and $j_1 \leq j \leq j_2$ will also contain $v_{l,t}$. In addition, since $\pi_f(i_1,j_1) > 0$ and $\pi_f(i_2,j_2) > 0$, $\pi_f(i,j) = 0$ for any (i,j)

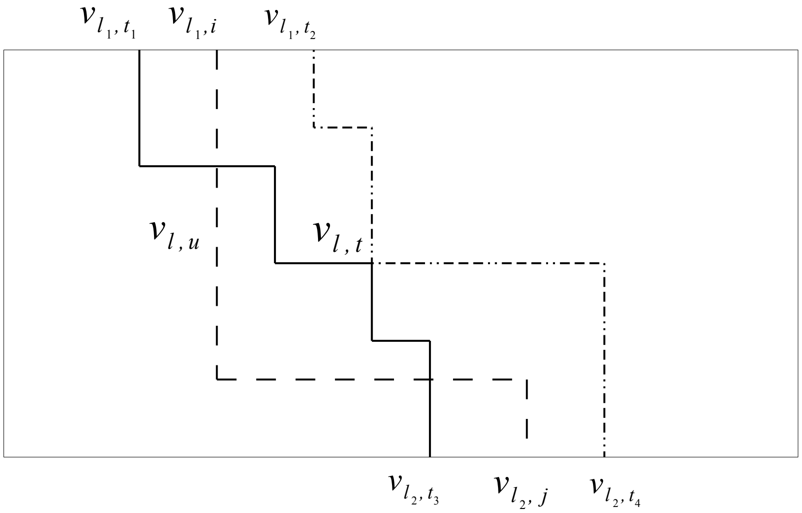


Figure 14: The cycle created in G_f if the path from $v_{l_1,i}$ to $v_{l_2,j}$ bypasses the node $v_{l,t}$.

pair such that $i > i_1, j < j_1$ or $i < i_2, j > j_2$. Therefore,

$$\begin{aligned}
\text{Inflow into } v_{l,t} &= \textstyle\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \pi_f(i,j) = \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \Delta_{i,j} \\
&= \textstyle\sum_{(i,j)\preceq(i_2,j_2)} \Delta_{i,j} - \sum_{(i,j)\prec(i_1,j_1)} \Delta_{i,j} \\
&= \min\{P_{i_2}, D_{j_2}\} - \max\{P_{i_1-1}, D_{j_1-1}\} \\
&= \gamma_2 - \gamma_1,
\end{aligned}$$

where $\gamma_1, \gamma_2 \in \Gamma$ and $\gamma_2 \geq \gamma_1$. The penultimate equality follows from the definition of $\Delta_{i,j}$. □

3.2.3 CFG with sources at echelon 0 and two echelons of sinks.

We consider the case of CFG where there are T sources at echelon 0 and two echelons of sinks, which we call CFG-2. CFG-2 generalizes the two-echelon ULS with intermediate demands in Zhang et al. [104] and two-echelon pure remanufacturing problems with intermediate demands. It is significantly harder to prove the polynomiality of the inflow values in CFG-2 than that of CFG-1, since there is no such invariant quantity π_f as in CFG-1. Our strategy is for each extreme flow in CFG-2 to calculate the inflow under a flow decomposition that satisfies certain properties similar to the properties of π_f in Proposition 2. In this way we are able to show that the inflow into any node can attain only a polynomial number of values under all extreme flows in CFG-2.

To simplify the analysis, we assume that in CFG-2 the sinks are at the last two echelons, echelon $L-1$ and echelon L. When sinks are at other two echelons, the flow decomposition needs to be adjusted accordingly, but the inflow calculation and the complexity result are not affected. Let the supply at source $v_{0,t}$ be p_t and the demand at sink $v_{l,t}$ be $d_{l,t}$ for $t \in \{1, \ldots, T\}$ and $l \in \{L-1, L\}$. Let $P_t = \sum_{i=1}^{t} p_i$ be the cumulative supply up to period t as in CFG-1 and $D_{l,t} = \sum_{i=1}^{t} d_{l,i}$ be the cumulative demand up to period t at echelon l for $l = L-1, L$. As in CFG-1, each flow decomposition for an extreme flow $\mathbf{f}$ can be represented by a vector whose components are the amount of flow sent along the paths between source-sink pairs. The difference is that we need *three* indices instead of two for a source-sink pair in CFG-2. Let $\mu_f(i, j, l)$ denote the amount of flow along the path from the source $v_{0,i}$ to the sink $v_{l,j}$ in G_f under some flow decomposition. Then each flow decomposition for $\mathbf{f}$ can be represented by a vector

$$\mu_f = (\mu_f(1,1,L), \mu_f(1,1,L-1), \mu_f(1,2,L), \mu_f(1,2,L-1), \ldots, \mu_f(T,T,L), \mu_f(T,T,L-1)).$$

Let χ_f be the lexicographically largest vector among all flow decomposition vectors for the extreme flow $\mathbf{f}$. Then χ_f satisfies some properties similar to those of π_f in CFG-1.

Proposition 5.

1. *For any $i_1 < i_2$, $j_1 < j_2$ and $l \in \{L-1, L\}$, $\chi_f(i_1, j_2, l) \cdot \chi_f(i_2, j_1, l) = 0$.*
2. *If $\chi_f(i_1, j_1, l) > 0$ and $\chi_f(i_1, j_2, l) > 0$ with $j_1 < j_2 - 1$ and $l \in \{L-1, L\}$, then $\chi_f(i_1, j, l) = d_{l,j}$ for any $j \in \{j_1 + 1, \ldots, j_2 - 1\}$.*

Proof. Similar to the proof of Proposition 2. □ □

Note that there is no similar result in CFG-2 to the statement 3 of Proposition 2. Given a sink v_{L,j_1}, even if $\chi_f(i_1, j_1, L), \chi_f(i_2, j_1, L) > 0$ with $i_1 < i_2$, $\chi_f(i, j_1, L)$ can be 0 instead of p_i for $i_1 < i < i_2$, since the source $v_{0,i}$ can satisfy demand only on echelon $L-1$. Let

$$\begin{aligned} X_L &= \cup_j \{D_{L,j}\} \cup \cup_{i,j} \{P_i - D_{L-1,j}\}, \\ X_{L-1} &= \cup_j \{D_{L-1,j}\} \cup \cup_{i,j} \{P_i - D_{L,j}\} \cup \cup_{i,j,k} \{P_i + D_{L-1,j} - P_k\}. \end{aligned} \tag{12}$$

Proposition 6. *For CFG-2, the inflow into any node under any extreme flow is*

$$\sum_{m=L-1}^{L} (\gamma_{m,2} - \gamma_{m,1}),$$

where $\gamma_{m,1}, \gamma_{m,2} \in X_m$ *with* $\gamma_{m,2} \geq \gamma_{m,1}$ *and* $m \in \{L-1, L\}$*.*

Proof. Given an extreme flow $\mathbf{f}$, the inflow into $v_{l,t}$ can be calculated under the flow decomposition χ_f as a summation of flows along paths that contain $v_{l,t}$. Let (i_1, j_1, L) and (i_2, j_2, L) be the lexicographically smallest and largest (i, j, L) tuples such that $\chi_f(i, j, L) > 0$ and the path from source $v_{0,i}$ to sink $v_{L,j}$ contains $v_{l,t}$ in G_f. Let $(i_3, j_3, L-1)$ and $(i_4, j_4, L-1)$ be the lexicographically smallest and largest $(i, j, L-1)$ tuples such that $\chi_f(i, j, L-1) > 0$ and the path from source $v_{0,i}$ to sink $v_{L-1,j}$ contains $v_{l,t}$ in G_f. Then by an argument similar to the one used in the proof of Theorem 6,

$$\begin{aligned}
\text{Inflow into } v_{l,t} &= \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \chi_f(i,j,L) + \sum_{i=i_3}^{i_4} \sum_{j=j_3}^{j_4} \chi_f(i,j,L-1) \\
&= \left[\sum_{(i,j,L) \preceq (i_2,j_2,L)} \chi_f(i,j,L) - \sum_{(i,j,L) \prec (i_1,j_1,L)} \chi_f(i,j,L) \right] + \\
&\quad \left[\sum_{(i,j,L) \preceq (i_4,j_4,L-1)} \chi_f(i,j,L-1) - \sum_{(i,j,L-1) \prec (i_3,j_3,L-1)} \chi_f(i,j,L-1) \right]
\end{aligned}$$

It remains to show that the term $\sum_{(i,j,m) \preceq (i',j',m)} \chi_f(i,j,m) \in X_m$ for any i', j' and $m = L-1, L$. The proof is based on induction on i'.

The base case $i' = 1$. Since $\sum_{(i,j,m) \preceq (1,j',m)} \chi_f(i,j,m) = \sum_{j=1}^{j'} \chi_f(1,j,m)$ for $m \in \{L-1, L\}$, we have to show that $\sum_{j=1}^{j'} \chi_f(1,j,L-1) \in X_{L-1}$ and $\sum_{j=1}^{j'} \chi_f(1,j,L) \in X_L$ for any j' under any extreme flow $\mathbf{f}$. First fix the extreme flow $\mathbf{f}$, let α be the largest time index j such that $\chi_f(1,j,L-1) > 0$ and β be the largest time index j such that $\chi_f(1,j,L) > 0$. Then $\sum_{j=1}^{j'} \chi_f(1,j,L-1) = \sum_{j=1}^{j'} d_{L-1,j} = D_{L-1,j'}$ for any $j' < \alpha$ and $\sum_{j=1}^{j'} \chi_f(1,j,L) = \sum_{j=1}^{j'} d_{L,j} = D_{L,j'}$ for any $j' < \beta$, according to statement 2 in Proposition 5. It remains to show that $\sum_{j=1}^{\alpha} \chi_f(1,j,L-1) \in X_{L-1}$ and $\sum_{j=1}^{\beta} \chi_f(1,j,L) \in X_L$. By the flow balance constraint at node $v_{0,1}$, we have

$$p_1 = \sum_{j=1}^{\alpha} \chi_f(1,j,L-1) + \sum_{j=1}^{\beta} \chi_f(1,j,L). \tag{13}$$

If either $\sum_{j=1}^{\alpha} \chi_f(1,j,L-1) = D_{L-1,\alpha}$ or $\sum_{j=1}^{\beta} \chi_f(1,j,L) = D_{L,\beta}$, then we are done. Otherwise we must have $\sum_{j=1}^{\alpha} \chi_f(1,j,L-1) < D_{L-1,\alpha}$ and $\sum_{j=1}^{\beta} \chi_f(1,j,L) < D_{L,\beta}$, implying that both the demand $d_{L-1,\alpha}$ and demand $d_{L,\beta}$ are only partially satisfied by the supply p_1. Let k be the smallest time index $i > 1$ such that $\chi_f(i,\beta,L) > 0$. It implies that under the flow decomposition χ_f, supply p_k is the first supply after p_1 to satisfy the demand $D_{L,\beta}$, and sources $v_{0,2}, \ldots, v_{0,k-1}$ make no contribution to demands at echelon L, as shown in Figure 15.

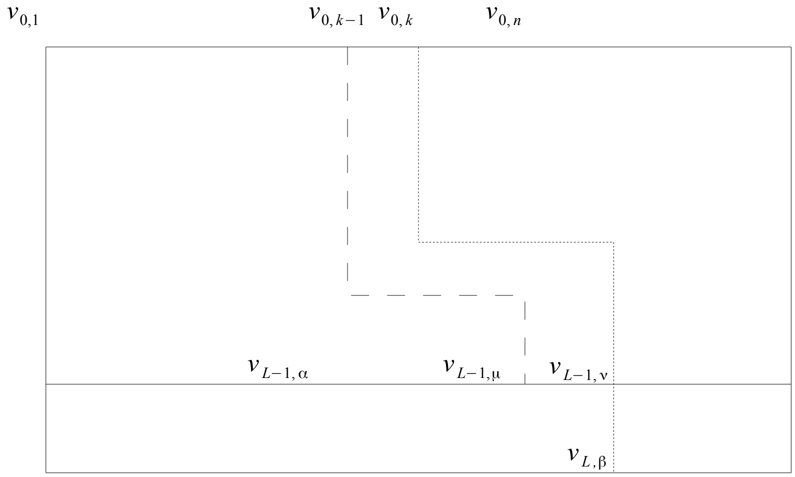


Figure 15: The case that $d_{L-1,\alpha}$ and $d_{L,\beta}$ are partially satisfied by p_1.

Claim There exists some $i_1 \in \{2, \ldots, k-1\}$ and $j_1 \in \{1, \ldots, T\}$ such that

$$\sum_{(i,j,L-1) \preceq (i_1,T,L-1)} \chi_f(i,j,L-1) = D_{L-1,j_1}.$$

The claim indicates that under the flow decomposition χ_f, supply $p_1, \ldots, p_{i_1}$ are decomposed to satisfy demand $d_{L-1,1}, \ldots, d_{L-1,j_1}$ at echelon $L-1$, demand $d_{L,1}, \ldots, d_{L,\beta-1}$ at echelon L and part of demand $d_{L,\beta}$. In addition, $p_2, \ldots, p_{i_1}$ only satisfy the demand at echelon $L-1$. Then

$$P_{i_1} = \sum_{(i,j,L-1) \preceq (i_1,T,L-1)} \chi_f(i,j,L-1) + \sum_{j=1}^{\beta} \chi_f(1,j,L). \tag{14}$$

By the claim and (14), $\sum_{j=1}^{\beta} \chi_f(1,j,L) = P_{i_1} - D_{L-1,j_1} \in X_L$. Then by (13),

$$\sum_{j=1}^{\alpha} \chi_f(1,j,L-1) = p_1 + D_{L-1,j_1} - P_{i_1} \in X_{L-1}.$$

The proof of the claim is based on contradiction. Suppose that the claim does not hold, then we show that there will be a cycle in the underlying undirected graph of G_f. If such i_1 and j_1 in the claim do not exist, then under the flow decomposition χ_f each supply p_i $(2 \leq i \leq k-1)$ ends up only fulfilling part of some demand on echelon $L-1$, which implies that $v_{0,i}$ is connected to $v_{0,i+1}$ through some sink on echelon $L-1$ since they both contribute to the demand at that sink. The source $v_{0,1}$ is connected to $v_{0,2}$ through the sink $v_{L-1,\alpha}$ by the same reason. In addition, $v_{0,k-1}$ is connected to $v_{0,k}$ through some sink on echelon $L-1$ as well due to the following reason. As shown in Figure 15, let μ be the largest time index j such that $\chi_f(k-1, \mu, L-1) > 0$, which implies that $v_{0,k-1}$ is connected to $v_{L-1,\mu}$ in G_f and there exists $n > k-1$ such that $\chi_f(n, \mu, L-1) > 0$. If $n = k$, then $v_{0,k-1}$ and $v_{0,k}$ are connected through the sink $v_{L-1,\mu}$. If $n > k$, then the path from $v_{0,k}$ to $v_{L,\beta}$ will intersect with either the path from $v_{0,k-1}$ to $v_{L-1,\mu}$ or the path from $v_{0,n}$ to $v_{L-1,\mu}$. Either case contradicts statement 1 of Proposition 5. Now $v_{0,1}$ is connected to $v_{0,k}$ through two different undirected paths, one through nodes $v_{0,2}, \ldots, v_{0,k-1}$ and sinks at echelon $L-1$, the other through the sink $v_{L,\beta}$ at echelon L. Thus there is a cycle containing $v_{0,1}$ and $v_{0,k}$ in the underlying undirected graph of G_f. We proved here the claim.

The induction step. Suppose that $\sum_{(i,j,m) \preceq (i',j',m)} \chi_f(i,j,m) \in X_m$ under all extreme flows for any $i' \leq q$. We want to show that $\sum_{(i,j,m) \preceq (q+1,j',m)} \chi_f(i,j,m) \in X_m$ for each j' and $m \in \{L-1, L\}$. The proof is similar as in the base case. Let α be the largest time index j such that $\chi_f(q+1, i, L-1) > 0$ and β be the largest time index j such that $\chi_f(q+1, j, L) > 0$.

We first show that $\sum_{(i,j,L-1) \preceq (q+1,j',L-1)} \chi_f(i,j,L-1) \in X_{L-1}$ each $j' < \alpha$.

1. If α is the only time index j such that $\chi_f(q+1, j, L-1) > 0$, which indicates that the supply p_{q+1} only contributes to demand $d_{L-1,\alpha}$ on echelon $L-1$, then for any $j' < \alpha$, $\sum_{(i,j,L-1) \preceq (q+1,j',L-1)} \chi_f(i,j,L-1) = \sum_{(i,j,L-1) \preceq (q,T,L-1)} \chi_f(i,j,L-1) \in X_{L-1}$.

2. If there are at least two j's such that $\chi_f(q+1, j, L-1) > 0$, then for each $j' < \alpha$, $\sum_{(i,j,L-1) \preceq (q+1,j',L-1)} \chi_f(i,j,L-1)$ either equals $\sum_{(i,j,L-1) \preceq (q,T,L-1)} \chi_f(i,j,L-1)$ or D_{L-1,j_1} for some j_1 on echelon $L-1$.

Similarly, $\sum_{(i,j,L)\preceq(q+1,j',L)} \chi_f(i,j,L) \in X_L$ for each $j' < \beta$.

Now it remains to show that the result holds for both $\sum_{(i,j,L-1)\preceq(q+1,\alpha,L-1)} \chi_f(i,j,L-1)$ and $\sum_{(i,j,L)\preceq(q+1,\beta,L)} \chi_f(i,j,L)$. By the flow balance constraint,

$$P_{q+1} = \sum_{(i,j,L-1)\preceq(q+1,\alpha,L-1)} \chi_f(i,j,L-1) + \sum_{(i,j,L)\preceq(q+1,\beta,L)} \chi_f(i,j,L). \tag{15}$$

If either $\sum_{(i,j,L-1)\preceq(q+1,\alpha,L-1)} \chi_f(i,j,L-1) = D_{L-1,\alpha}$ or $\sum_{(i,j,L)\preceq(q+1,\beta,L)} \chi_f(1,j,L) = D_{L,\beta}$, then we are done. Otherwise the demand $d_{L-1,\alpha}$ and $d_{L,\beta}$ is partially satisfied by the supply up to period $q+1$. By an argument similar to the one in the base case, we can show that $\sum_{(i,j,L-1)\preceq(q+1,\alpha,L-1)} \chi_f(i,j,L-1) \in X_{L-1}$ and $\sum_{(i,j,L)\preceq(q+1,\beta,L)} \chi_f(1,j,L) \in X_L$. □

Theorem 8. *For fixed L, CFG-2 can be solved in polynomial time in T and the number of queries of a function-value oracle.*

Proof. By (12), the cardinality of X_{L-1} is $O(T^3)$ and the cardinality of X_L is $O(T^2)$. By Proposition 6, the inflow into any node is a summation of the difference of two elements in X_{L-1} and the difference of two elements in X_L, so it can attain $O(T^{3+3} \cdot T^{2+2}) = O(T^{10})$ values under all extreme flows in CFG-2. Then in the DP formulation for CFG-2, the cardinality of the state space at each stage is $O(T^{10L})$. Thus CFG-2 can be solved in polynomial time by solving a shortest path problem over an acyclic network. □

3.3 CFG with L echelons of sinks

After the results in section 3.2 were submitted as a paper to *Mathematical Programming*, we discovered more general results that subsume the results already given, but for the sake of completeness we decide to keep the original proofs in section 3.2 as well as give the new results and proofs. In this section, we show that CFG with L echelons of sinks is polynomially solvable with fixed L and NP-hard when L is an input parameter. We first provide a new DP framework with the component of the state variable being the flow over some horizontal arc, and then show that the cardinality of the state space is polynomial in T based on the path decomposition of the extreme flow.

3.3.1 The new DP framework.

We propose to solve CFG by using a new discrete time DP formulation. The elements of the DP are as follows.

1. Decision stages. There are $T+1$ stages which corresponds to time period $t = 0, 1, \ldots, T$.

2. States. Define the state $\mathbf{s}^t$ at stage t to be a $(L+1)$-dimensional vector whose component s_l^t denotes the flow over the forward arc $(v_{l,t}, v_{l,t+1})$, or in practice the inventory level at echelon l and time period t. We assume that each component of $\mathbf{s}^0$ and $\mathbf{s}^T$ is 0. Note that the dimension of $\mathbf{s}^t$ can be reduced by one since the summation of the components of $\mathbf{s}^t$ is always $\sum_{l=0}^{L+1} b(v_{l,t})$ by flow balance constraints.

3. Decision variables (or actions). The decision variable $\mathbf{u}^t$ at stage t is a L-dimensional vector whose component u_l^t denotes the flow over the downward arc $(v_{l,t+1}, v_{l+1,t+1})$, or in practice the production level at echelon l and time period $t+1$.

4. The system equations. The state $\mathbf{s}^{t+1}$ at stage $t+1$ can be easily calculated by the flow balance constraints of the nodes at time period $t+1$. Let the system equations be $\mathbf{s}^{t+1} = H_t(\mathbf{s}^t, \mathbf{u}^t)$, where H_t is the affine function representing the flow balance constraints for nodes at stage $t+1$.

5. The cost function. The cost at stage t is the sum of all costs incurred by the downward arcs and forward arcs at that stage, or the sum of holding costs at the end of period t and production costs at period $t+1$. Let $r_t(\mathbf{s}^t, \mathbf{u}^t)$ denote the cost incurred at stage t.

Then CFG is formulated as a discrete time DP problem with the linear system $\mathbf{s}^{t+1} = H_t(\mathbf{s}^t, \mathbf{u}^t)$ and cost function r_t over $T+1$ stages. This DP formulation is also difficult to solve directly, since the state space at stage t is an uncountable set in general. However by a similar argument to that in Section 3.2, it suffices to consider those states corresponding to the extreme points of P_F, the number of which is finite. To argue that this DP formulation can be solved in polynomial time, it remains to show that the cardinality of the state space at each stage is polynomial in T. Since the dimension of the state vector at each stage is $L+1$, the task is reduced to show that each component of the state vector, namely the flow over each forward arc under all extreme points of P_F, can take on a finite set of values whose cardinality is polynomial in T. We first present a result that can be seen as the "arc" version of Proposition 4.

Proposition 7. *Given any extreme flow* $\mathbf{f}$*, let* $\mathcal{Q}_1$ *be a path from* v_{l_1,t_1} *to* v_{l_2,t_3} *and* $\mathcal{Q}_2$ *be a path from* v_{l_1,t_2} *to* v_{l_2,t_4} *in* G_f *with* $l_1 < l_2, t_1 \leq t_2$ *and* $t_3 \leq t_4$*. If* $\mathcal{Q}_1$ *and* $\mathcal{Q}_2$ *both contain arc* a*, then any path from* $v_{l_1,i}$ *to* $v_{l_2,j}$ *in* G_f *with* $t_1 \leq i \leq t_2$ *and* $t_3 \leq j \leq t_4$ *also contains the arc* a*.*

Proof. Proof by contradiction. As shown in Figure 16, the arc $a = (v_{l,t}, v_{l,t+1})$ is contained in the path from v_{l_1,t_1} to v_{l_2,t_3} and the path from v_{l_1,t_2} to v_{l_2,t_4} in G_f. Suppose that there exists some pair (i,j) with $t_1 \leq i \leq t_2$ and $t_3 \leq j \leq t_4$ such that the path from $v_{l_1,i}$ to $v_{l_2,j}$ bypasses the arc a. Then the path must contain some node $v_{l,u}$ with either $u \leq t$ or $u \geq t+1$. If $u \leq t$, by Observation 1, the path from v_{l_1,t_1} to $v_{l,t}$ must intersect with the path from $v_{l_1,i}$ to $v_{l,u}$, and the path from $v_{l,t+1}$ to v_{l_2,t_3} must intersect with the path $v_{l,u}$ to $v_{l_2,j}$ in G_f. The two intersections create a cycle in the underlying undirected graph of G_f, a contradiction. The argument is essentially the same if $u \geq t+1$.

□

3.3.2 CFG with sources at echelon 0 and L echelons of sinks.

Now we consider the general case of CFG where sources are at echelon 0 and sinks are at echelon 1 to echelon L. CFG generalizes the two-echelon ULS with intermediate demands in Zhang et al. [104] and multi-echelon pure remanufacturing problems with intermediate demands. Our proof strategy is for each extreme flow in CFG to calculate the flow over each arc under a flow decomposition that satisfies certain properties similar to the properties of π_f in Proposition 2. In this way we are able to show that the flow over each forward arc can attain only a polynomial number of values under all extreme flows in CFG.

Let the supply at source $v_{0,t}$ be p_t and the demand at sink $v_{l,t}$ be $d_{l,t}$ for $t \in \{1, \ldots, T\}$

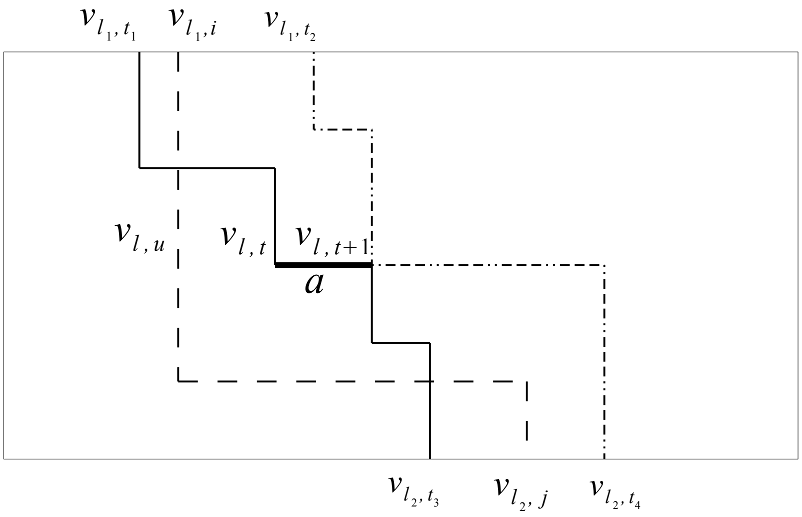


Figure 16: The cycle created in G_f if the path from $v_{l_1,i}$ to $v_{l_2,j}$ bypasses the arc a.

and $l \in \{1, \ldots, L\}$. Let $P_t = \sum_{i=1}^{t} p_i$ be the cumulative supply up to period t and $D_{l,t} = \sum_{i=1}^{t} d_{l,i}$ be the cumulative demand up to period t at echelon l for $l \in \{1, \ldots, L\}$. Given an extreme flow f, let $\mu_f(i, j, l)$ denote the amount of flow along the path from the source $v_{0,i}$ to the sink $v_{l,j}$ in G_f under some flow decomposition. Then each flow decomposition for $\mathbf{f}$ can be represented by a vector

$$\begin{aligned}\mu_f =&(\mu_f(1,1,L), \mu_f(1,1,L-1), \ldots, \mu_f(1,1,1),\\ &\mu_f(1,2,L), \mu_f(1,2,L-1), \ldots, \mu_f(1,2,1), \ldots,\\ &\mu_f(T,T,L), \mu_f(T,T,L-1), \ldots, \mu_f(T,T,1)).\end{aligned}$$

Let χ_f be the lexicographically largest vector among all flow decomposition vectors for the extreme flow $\mathbf{f}$. Then χ_f satisfies some properties similar to those of π_f in CFG-1.

Proposition 8.

1. *For any $i_1 < i_2$, $j_1 < j_2$ and $l \in \{1, \ldots, L\}$, $\chi_f(i_1, j_2, l) \cdot \chi_f(i_2, j_1, l) = 0$.*

2. *If $\chi_f(i_1, j_1, l) > 0$ and $\chi_f(i_1, j_2, l) > 0$ with $j_1 < j_2 - 1$, then $\chi_f(i_1, j, l) = d_{l,j}$ for any $j \in \{j_1 + 1, \ldots, j_2 - 1\}$ and $l \in \{1, \ldots, L\}$.*

Note that there is no similar result to the statement 3 of Proposition 2. Given a sink v_{L,j_1}, even if $\chi_f(i_1, j_1, L), \chi_f(i_2, j_1, L) > 0$ with $i_1 < i_2$, $\chi_f(i, j_1, L)$ can be 0 instead of p_i for $i_1 < i < i_2$, since the source $v_{0,i}$ can satisfy demand only at echelon 1 to echelon $L - 1$.

Our main result in this section is as follows.

Proposition 9. *For fixed L, CFG can be solved in polynomial time in T and the number of queries of a function-value oracle.*

The result follows from the proposition below and the DP algorithm proposed in this section.

Proposition 10. *For fixed L, the number of values that the flow over each arc can attain is polynomial in T under all extreme flows of CFG.*

Proof. Given an extreme flow $\mathbf{f}$, the inflow into any arc $a \in A$ can be calculated under the flow decomposition χ_f as a summation of flows along paths that contain arc a. Let (i_b^l, j_b^l, l) and (i_e^l, j_e^l, l) be the lexicographically smallest and largest (i, j, l) tuples such that the path from source $v_{0,i}$ to sink $v_{l,j}$ contains arc a in G_f and $\chi_f(i, j, l) > 0$. Without loss of generality, we assume that arc a is a forward arc and $a = (v_{m,t}, v_{m,t+1})$. Then by an argument similar to the one used in the proof of Theorem 6, the flow over arc a is

$$\begin{aligned} f_a =& \sum_{l=m}^{L} \sum_{(i_b^l, j_b^l) \preceq (i,j) \preceq (i_e^l, j_e^l)} \chi_f(i, j, l) \\ =& \sum_{l=m}^{L} \left[\sum_{(i,j) \preceq (i_e^l, j_e^l)} \chi_f(i, j, l) - \sum_{(i,j) \prec (i_b^l, j_b^l)} \chi_f(i, j, l) \right] \end{aligned} \tag{16}$$

If the following claim holds, then with equation (16) we are able to prove that f_a can only attain a polynomial number of values in T under all extreme flows for fixed L.

Claim 1 For fixed L, the number of values that $\sum_{(i,j) \preceq (i',j')} \chi_f(i, j, l)$ can attain is polynomial in T for any $i', j' \in \{1, \ldots, T\}$, $l \in \{1, \ldots, L\}$ under all extreme flows.

Proof of Claim 1 We prove the claim by applying induction on the time index i' . First we prove the base case $i' = 1$.

For $i' = 1$, we have $\sum_{(i,j) \preceq (i',j')} \chi_f(i, j, l) = \sum_{j=1}^{j'} \chi_f(1, j, l)$. We need to show that the number of values that $\sum_{j=1}^{j'} \chi_f(1, j, l)$ can attain is polynomial in T for any j' and l. We will prove this by induction on l.

1. The base case $l = L$. Our goal is to show that

$$\sum_{j=1}^{j'} \chi_f(1, j, L) \in Y = \{D_{L,w} | 1 \le w \le T\} \cup \{P_{i_u} - \sum_{l=1}^{L-1} D_{l,u_l} | 1 \le i_u, u_1, \ldots, u_{L-1} \le T\}$$

for any j' under any extreme flow $\mathbf{f}$. Since the cardinality of Y is $O(T^L)$, then $\sum_{j=1}^{j'} \chi_f(1,j,L)$ can attain a polynomial number of values for any j' under all extreme flows.

Fix the extreme flow $\mathbf{f}$, let $j_{1,L} = \arg\max\{j|\chi_f(1,j,L) > 0\}$. Then at echelon L, source $v_{0,1}$ contributes to sinks up to time period $j_{1,L}$. By Proposition 8, $\chi_f(1,j,L) = d_{L,j}$ for $j < j_{1,L}$. Then

$$\sum_{j=1}^{j'} \chi_f(1,j,L) = \begin{cases} D_{L,j'}, & j' < j_{1,L} \\ \sum_{j=1}^{j_{1,L}} \chi_f(1,j,L), & j' \geq j_{1,L}. \end{cases}$$

If $\chi_f(1,j_{1,L},L) = d_{L,j_{1,L}}$ for the extreme flow $\mathbf{f}$, then $\sum_{j=1}^{j'} \chi_f(1,j,L) = D_{L,j_{1,L}}$ for all $j' \geq j_{1,L}$. In this case, the number of values that $\sum_{j=1}^{j'} \chi_f(1,j,L)$ can attain is $O(T)$ for all possible j's.

It only remains to show that $\sum_{j=1}^{j_{1,L}} \chi_f(1,j,L)$ can only attain values in set Y when $0 < \chi_f(1,j_{1,L},L) < d_{L,j_{1,L}}$. Since $\chi_f(1,j_{1,L},L) < d_{L,j_{1,L}}$, the demand $d_{L,j_{1,L}}$ needs to be partially satisfied by sources in later time period. Let v_{0,i_r} be such source with i_r smallest, so i_r is the smallest time index $i > 1$ such that $\chi_f(i,j_{1,L},L) > 0$. Then we have $\chi_f(i,j,L) = 0$ for any $1 < i < i_r$, and the sources $v_{0,2}, v_{0,3}, \ldots, v_{0,i_r-1}$ have no contribution to any sink at echelon L. We will show the following claim holds.

Claim 2 Given any extreme flow $\mathbf{f}$, if $\chi_f(1,j_{1,L},L) < d_{L,j_{1,L}}$, there exists a time index $i_u < i_r$ such that

$$\sum_{j=1}^{j_{1,L}} \chi_f(1,j,L) = P_{i_u} - \sum_{l=1}^{L-1} D_{l,u_l}$$

for some $u_1, \ldots, u_{L-1} \in \{1, \ldots, T\}$.

Proof of Claim 2. Let

$$j_{i,l} = \arg\max\{j|\chi_f(i,j,l) > 0\}$$

be the largest time index j such that $\chi_f(i,j,l) > 0$ (set $j_{i,l} = 0$ if $\chi_f(i,j,l) = 0$ for each j). Since sources $v_{0,2}, \ldots, v_{0,i_r-1}$ only contribute to sinks at echelon 1 to echelon $L-1$, by summing up the flow balance constraints for sources $v_{0,1}, v_{0,2}, \ldots, v_{0,i}$ we have

$$P_i = \sum_{j=1}^{j_{1,L}} \chi_f(1,j,L) + \sum_{l=1}^{L-1} \sum_{(i,j)\preceq(i,j_{i,l})} \chi_f(i,j,l),$$

for each $i < i_r$. Then $\sum_{j=1}^{j_{1,L}} \chi_f(1,j,L) = P_i - \sum_{l=1}^{L-1} \sum_{(i,j)\preceq(i,j_{i,l})} \chi_f(i,j,l)$. We prove Claim 2 by contradiction. Suppose that the claim is not true, then we must have for each $k \in \{1,\ldots,i_r-1\}$, there exists at least one $j_{k,l}$ such that $D_{l,j_{k,l}-1} < \sum_{(i,j)\preceq(k,j_{k,l})} \chi_f(i,j,l) < D_{l,j_{k,l}}$. In this case there will be two distinctive paths from source $v_{0,1}$ to sink v_{0,i_r} in the underlying undirected graph of G_f, contradicting to the fact that the underlying undirected graph of G_f is acyclic. (In the rest of the proof, by "path" we mean the undirected path in the underlying undirected graph of G_f.) The first path is the path from $v_{0,1}$ to $v_{L,j_{1,L}}$ since $\chi_f(1,j_{1,L},L) > 0$, concatenated by the path from v_{0,i_r} to $v_{L,j_{1,L}}$ since $\chi_f(i_r,j_{1,L},L) > 0$. The second path does not contain any node at echelon L, therefore different from the first one. The second path is constructed as follows.

By assumption, there exists at least one j_{1,l_1} for some $l_1 \in \{1,\ldots,L-1\}$ such that $D_{l_1,j_{1,l_1}-1} < \sum_{(i,j)\preceq(1,j_{1,l_1})} \chi_f(i,j,l_1) < D_{l_1,j_{1,l_1}}$. Choose such a j_{1,l_1} with the echelon index l_1 being the largest, as shown in Figure 17. Since $\chi_f(1,j_{1,l_1},l_1) > 0$ by the definition of j_{1,l_1}, there is a path from $v_{0,1}$ to $v_{l_1,j_{1,l_1}}$. Since $\chi_f(1,j_{1,l_1},l_1) < d_{l_1,j_{1,l_1}}$, there exists a source v_{0,i_2} with $i_2 > 1$ such that $\chi_f(i_2,j_{1,l_1},l_1) > 0$. Choose such a v_{0,i_2} with i_2 being the smallest, so sources $v_{0,2}, v_{0,3}, \ldots, v_{0,i_2-1}$ only contribute to sinks at echelon 1 to echelon $l_1 - 1$. Then $v_{l_1,j_{1,l_1}}$ is connected to v_{0,i_2}.

If $i_2 \geq i_r$, it is not difficult to verify that the path from $v_{0,1}$ to v_{0,i_2} intersects with the path from v_{0,i_r} to $v_{L,j_{1,L}}$, so we can construct a path from $v_{0,1}$ to v_{0,i_r} containing only nodes at echelon 0 to echelon l_1. If $i_2 < i_r$, by assumption, there exists a j_{i_2,l_2} for some $l_2 \in \{1,\ldots,L-1\}$ such that $D_{l_2,j_{i_2,l_2}-1} < \sum_{(i,j)\preceq(i_2,j_{i_2,l_2})} \chi_f(i,j,l_2) < D_{l_2,j_{i_2,l_2}}$. Choose such a j_{i_2,l_2} with l_2 being the largest. We would like to show that there is path from v_{0,i_2} to $v_{l_2,j_{i_2,l_2}}$.

(a) Case $l_1 \leq l_2$. We have $\sum_{(i,j)\preceq(1,j_{1,l_2})} \chi_f(1,j,l_2)$ equals to some cumulative demand at echelon l_2 due to the choice of l_1, and $\sum_{(i,j)\preceq(i_2,j_{i_2,l_2})} \chi_f(i,j,l_2)$ is

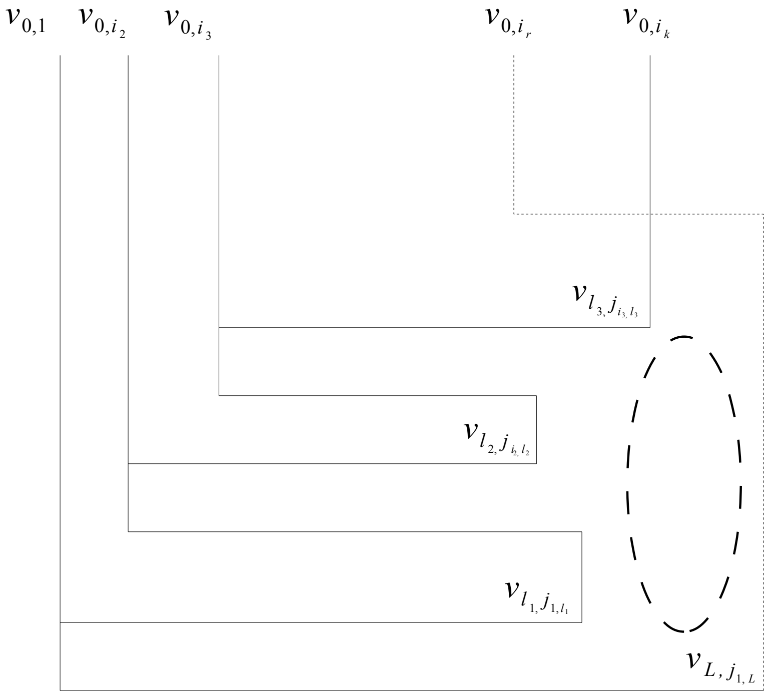


Figure 17: The circle that traverses nodes $v_{L,j_{1,L}}$,$v_{l_1,j_{1,l_2}}$, $v_{l_2,j_{i_2,l_2}}$ and $v_{l_3,j_{i_3,l_3}}$.

not equal to any cumulative demand at echelon l_2. Meanwhile, since sources $v_{0,2},\dots,v_{0,i_2-1}$ make no contribution to the sinks at echelon l_1, they make no contribution to sinks at echelon l_2, either. Then we must have $\chi_f(i_2, j_{i_2,l_2}, l_2) > 0$, and there is a path from v_{0,i_2} to $v_{l_2,j_{i_2,l_2}}$.

(b) Case $l_1 > l_2$. If $\chi_f(i_2, j_{i_2,l_2}, l_2) = 0$, by assumption $\sum_{(i,j)\preceq(i_2,j_{i_2,l_2})} \chi_f(i,j,l_2)$ is not equal to any cumulative demand $D_{l_2,j}$, then there must exist i'_1, i'_2 such that $i'_1 < i_2 < i'_2$, $\chi_f(i'_1, j_{i_2,l_2}, l_2) > 0$ and $\chi_f(i'_2, j_{i_2,l_2}, l_2) > 0$. Then there is one path from v_{0,i'_1} to $v_{l_2,j_{i_2,l_2}}$ and another path from v_{0,i'_2} to $v_{l_2,j_{i_2,l_2}}$. One of them needs to intersect the path from v_{0,i_2} to $v_{l_1,j_{1,l_1}}$. Therefore, there is a path from v_{0,i_2} to $v_{l_2,j_{i_2,l_2}}$.

Since the demand at the sink $v_{l_2,j_{i_2,l_2}}$ is not fulfilled by sources $v_{0,1},\dots,v_{0,i_2}$, there must exist a source v_{0,i_3} with $i_3 > i_2$ such that $\chi_f(i_3, j_{i_2,l_2}, l_2) > 0$. If $i_3 \geq i_r$, then the path from $v_{0,1}$ to v_{0,i_3} must intersect with the path from v_{0,i_r} to $v_{L,j_{1,L}}$, so we can construct a path from $v_{0,1}$ to v_{0,i_r} containing only nodes above echelon L. Otherwise if $i_3 < i_r$, there exists a j_{i_3,l_3} such that $\sum_{(i,j)\preceq(i_3,j_{i_3,l_3})} \chi_f(i,j,l_3)$ does not equal to any cumulative demand at echelon l_3 and there is also a path from v_{0,i_3} to $v_{l_3,j_{i_3,l_3}}$ by the similar argument. Continue this procedure until there is a source v_{0,i_k} with $i_k \geq i_r$, then we find a path from $v_{0,1}$ to v_{0,i_k} through $v_{l_1,j_{1,l_1}}, v_{0,i_2}, v_{l_2,j_{i_2,l_2}}, v_{0,i_3}, \ldots, v_{0,i_{k-1}}$ containing nodes only at echelon 0 to echelon $L-1$. This path must intersect with the path from v_{0,i_r} to $v_{L,j_{1,L}}$. Thus there is always a second path from $v_{0,1}$ to v_{0,i_r} which does not contain any node at echelon L, contradicting to that the underlying undirected graph of G_f is acyclic.

2. The induction step. Suppose that the number of values that $\sum_{j=1}^{j'} \chi_f(1,j,l)$ can attain is polynomial in T for all j' and $l > l'$, we would like to show that the result also holds for $l = l'$.

   (a) If $\chi_f(1, j_{1,l'}, l') = d_{l',j_{1,l'}}$, then

   $$\sum_{j=1}^{j'} \chi_f(1,j,l') = \begin{cases} D_{l',j'}, & j' < j_{1,l'} \\ D_{l',j_{1,l'}}, & j' \geq j_{1,l'}. \end{cases}$$

   (b) If $0 < \chi_f(1, j_{1,l'}, l') < d_{l',j_{1,l'}}$, then

   $$\sum_{j=1}^{j'} \chi_f(1,j,l') = \begin{cases} D_{l',j'}, & j' < j_{1,l'} \\ \sum_{j=1}^{j_{1,l'}} \chi_f(1,j,l'), & j' \geq j_{1,l'}. \end{cases}$$

   Let i'_r be the smallest index $i > 1$ such that $\chi_f(i, j_{1,l'}, l') > 0$, then we can show that there exists some $i'_u < i'_r$ and $u_1, u_2, \ldots, u_{l'-1} \in \{1, \ldots, T\}$ such that

   $$\sum_{j=1}^{j_{1,l'}} \chi_f(1,j,l') = P_{i'_u} - \sum_{l=1}^{l'-1} D_{l,u_l} - \sum_{l=l'+1}^{L} \sum_{j=1}^{j_{1,l}} \chi_f(1,j,l).$$

   The validity of the above equality follows from a similar argument to the proof of Claim 2. Then by the induction hypothesis, the number of values that $\sum_{j=1}^{j_{1,l'}} \chi_f(1,j,l')$ can attain is polynomial in T under all extreme flows.

So far we have finished the proof for the base case $i' = 1$. Now we proceed to the induction step for $i' = k$. Suppose that the number of values that $\sum_{(i,j)\preceq(i',j')} \chi_f(i,j,l)$ can attain is polynomial in T for all j', l and $i' < k$, we would like to show this is also the case for $i' = k$. The proof is based on induction on the echelon index l similar to the proof of the base case $i' = 1$. Consider the base case $l = L$. Fix an extreme flow $\mathbf{f}$, the result holds in all three cases below.

1. $\chi_f(k, j, L) = 0$ for all j. Or equivalently, the source $v_{0,k}$ makes no contribution to the sinks on echelon L. Then $\sum_{(i,j)\preceq(k,j')} \chi_f(i, j, L) = \sum_{(i,j)\preceq(k-1,T)} \chi_f(i, j, L)$, and the result follows from the induction hypothesis.

2. $\chi_f(k, j_{k,L}, L) > 0$ and $\sum_{(i,j)\preceq(k,j_{k,L})} \chi_f(i, j, L) = D_{L,j_{k,L}}$. Then

$$\sum_{(i,j)\preceq(k,j')} \chi_f(i,j,L) = \begin{cases} \sum_{(i,j)\preceq(k-1,T)} \chi_f(i,j,L), & j' < j_{k,L}, \chi_f(k,j',L) = 0 \\ D_{L,j'}, & j' < j_{k,L}, \chi_f(k,j',L) > 0 \\ D_{L,j_{k,L}}, & j' \geq j_{k,L}. \end{cases}$$

   Then the result follows from the induction hypothesis.

3. $\chi_f(k, j_{k,L}, L) > 0$ and $\sum_{(i,j)\preceq(k,j_{k,L})} \chi_f(i, j, L) < D_{L,j_{k,L}}$.

$$\sum_{(i,j)\preceq(k,j')} \chi_f(i,j,L) = \begin{cases} \sum_{(i,j)\preceq(k-1,T)} \chi_f(i,j,L), & j' < j_{k,L}, \chi_f(k,j',L) = 0 \\ D_{L,j'}, & j' < j_{k,L}, \chi_f(k,j',L) > 0 \\ \sum_{(i,j)\preceq(k,j_{k,L})} \chi_f(i,j,L), & j' \geq j_{k,L}. \end{cases}$$

   By a similar proof to that of Claim 2, there exists some $i_u, u_1, u_2, \ldots, u_{L-1} \in \{1, \ldots, T\}$, such that

$$\sum_{(i,j)\preceq(k,j_{k,L})} \chi_f(i,j,L) = P_{i_u} - \sum_{l=1}^{L-1} D_{l,u_l}.$$

The induction step for case $i' = k$ is similar to the proof of the induction step for the base case $i' = 1$. □

Note that the parameter L being fixed is a critical condition in Proposition 10. As shown in the following proposition, CFG is NP-hard if the number of echelons $(L + 1)$ is also an input parameter. Therefore, Proposition 10 is unlikely to hold without the condition that L is fixed unless P=NP.

Proposition 11. *CFG is NP-hard given the input L, T, supply vector b and the function-value oracle for the cost over each arc.*

Proof. We provide a polynomial-time reduction to CFG from the partition problem. An instance of the partition problem asks that given a set S of integers $y_1, \dots, y_n$ whether there exists a partition of S such that the sum of the numbers in each partition is equal to $\sum_{i=1}^{2} y_i/2$. We construct an instance of CFG and show that the minimum cost of that instance of CFG is n if and only if the partition instance is a yes instance.

Consider a grid network with $n+1$ echelons, $n+1$ periods, two sources and n sinks, as shown in Figure 18. Here $L = n$ and $T = n+1$. The two sources are $v_{0,1}$ and $v_{0,2}$ with $b(v_{0,1}) = b(v_{0,2}) = \sum_{i=1}^{n} y_i/2$. The n sinks are $v_{1,2}, v_{2,3}, \dots, v_{n,n+1}$ with $b(v_{i,i+1}) = -y_i$ for $i \in \{1, \dots, n\}$. The cost function over each incoming arc for the sinks $v_{i,i+1}$ is $\mathbb{I}(x_a > 0)$, i.e., the cost is 1 if the flow over the arc is strictly positive and 0 otherwise. The cost over each of the rest arcs present in Figure 18 is always 0, and the cost over each arc not present in Figure 18 is large enough so that the arc will never be used in any optimal flow (for example, the cost function is constant with value $2n$). The construction can be done in polynomial time. Then one can verify that the minimum cost of this instance is n if and only if the partition instance is a yes instance and the minimum cost is $n+1$ if and only if partition instance is a no instance. □

3.4 Extensions

We study two extensions of the CFG model. The first extension considers CFG with backward arcs between two consecutive time periods at each echelon. The backward arcs are used to model the option of backlogging in supply chain management. We call this extension CFG-B.

Proposition 12. *For fixed L, CFG-B can be solved in polynomial time in T and the number of queries of a function-value oracle.*

Proof. CFG-B can be formulated as a discrete-time DP with $T+1$ stages, similar to the one for CFG in Section 3.3. The only difference is that the state at stage t for CFG-B includes not only the flow over each forward arc between time period t and $t+1$, but also the flow over each backward arc between time period t and $t+1$. The dimension of the state becomes $2(L+1)$, which is still constant for fixed L. We can show that the number

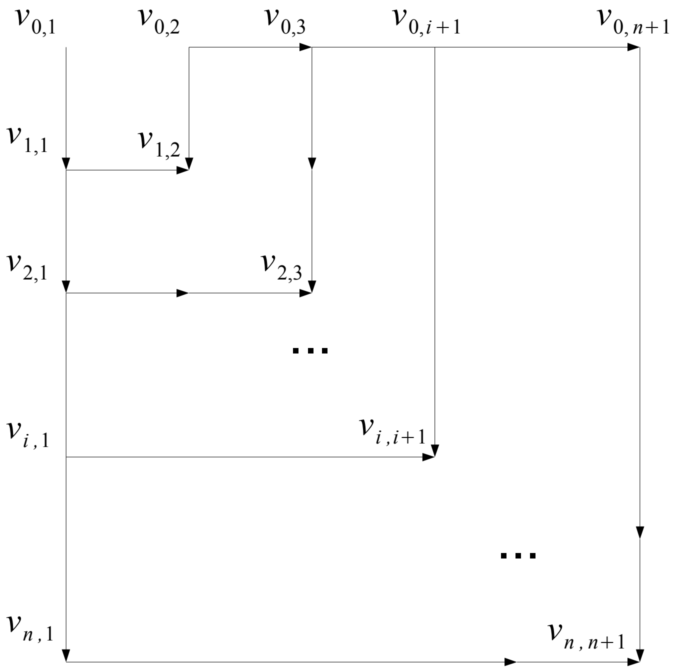


Figure 18: The instance of CFG.

of values that each horizontal arc can take under all extreme flows is polynomial in T by a similar argument to that for CFG in Section 3.3. Then the cardinality of the state space at each stage is polynomial in T, and CFG-B can be solved in polynomial time in T and the number of queries of a function-value oracle. □

The second extension considers CFG with upward arcs between two consecutive echelons at each time period. The upward arcs are used to model the return of used products in supply chain management. We call this extension CFG-U.

Proposition 13. *CFG-U with at least three echelons is NP-hard.*

Proof. We provide a polynomial-time reduction from the partition problem to CFG-U. An instance of the partition problem asks that given a set S of integers $y_1, \dots, y_n$ whether there exists a partition of S such that the sum of the numbers in each partition is equal to $\sum_{i=1}^{2} y_i/2$. We construct an instance of CFG-U and show that the minimum cost of that instance of CFG is n if and only if the partition instance is a yes instance.

Consider a grid network with $(L + 1)$ echelons $(L \geq 2)$, $n + 1$ periods, two sources and n sinks, as shown in Figure 19. Note that only echelon 0 to echelon 2 of the graph is shown. All nodes below echelon 2 are transshipment nodes, and the cost over each arc below echelon 2 is large enough so that the arc will never be used in any optimal flow (for example, the cost function is constant with value $2n$). The two sources are $v_{0,1}$ and $v_{0,2}$ with $b(v_{0,1}) = b(v_{0,2}) = \sum_{i=1}^{n} y_i/2$. The n sinks are $v_{1,2}, v_{1,3}, \ldots, v_{1,n+1}$ with $b(v_{1,i+1}) = -y_i$ for $i \in \{1, \ldots, n\}$. The cost function over each of the downward arc $a = (v_{0,t}, v_{1,t})$ and upward arc $a = (v_{2,t}, v_{1,t})$ for $t = 1, \ldots, n$ (the bold arcs in Figure 19) is $\mathbb{I}(x_a > 0)$, i.e., the cost is 1 if the flow over the arc is strictly positive and 0 otherwise. The cost over the downward arc $(v_{0,1}, v_{1,1})$ and upward $(v_{1,1}, v_{2,1})$ is always 0, the cost over each forward arc present in Figure 19 is always 0, and the cost over each arc not present in Figure 19 is large enough so that the arc will never be used in any optimal flow. The construction can be done in polynomial time. Then one can verify that the minimum cost of this instance is n if and only if the partition instance is a yes instance and the minimum cost is $n + 1$ if and only if partition instance is a no instance. □

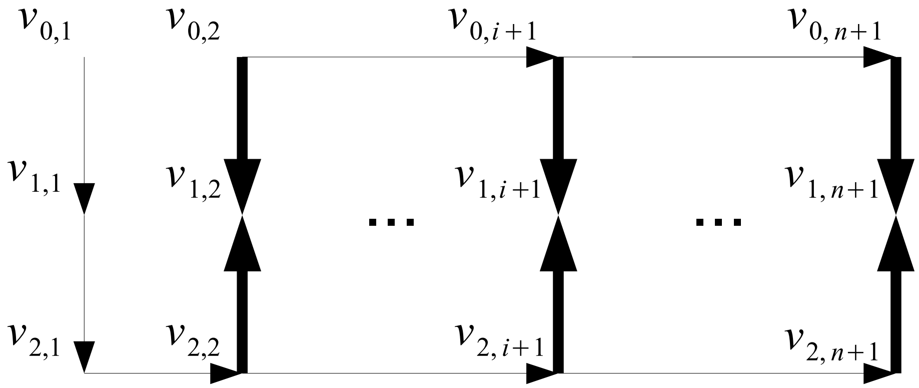


Figure 19: The instance of CFG-U.

3.5 Conclusions and future work

In this chapter, we studied the minimum concave cost flow problem over a grid network. We proposed a polynomial-time algorithm for the problem when the sources are at the first echelon and the sinks are at L echelons with L being fixed. Our result unifies the complexity results for the lot-sizing problem and several variants (multi-echelon, backlogging) in production planning and the pure remanufacturing problem in green recycling, and gives the first polynomial-time algorithm for some problems whose complexities were not known before. The main technical contribution is discovering a particular path decomposition (the lexicographically largest one), which provides a smart way to count the number of possible flow values over each arc under extreme flows and a unified framework to analyze the complexity of various lot-sizing models. We also showed that several variants of CFG are NP-hard, suggesting that the complexity of CFG depends on not only the underlying graph topology but also the arc directions and the distribution of sources and sinks.

There are certainly more questions left open than we have answered in this chapter. The study of MCCNFP can be further pursued in the following directions.

1. Computational complexity for other types of uncapcacited networks. Since the computational complexity of CFG depends not only on the graph structure but also the arc directions and distribution of sources and sinks, it will be interesting to see whether CFG with only forward and downward arcs and a fixed number of echelons is still polynomially solvable if we allow the distribution of sources and sinks to be arbitrary. Our conjecture is that it is indeed the case, and a positive answer will generalize the result of Proposition 9. It is also unknown if CFG with a single source is polynomially solvable when the number of echelons is an input parameter.

2. Capacitated networks. Another important direction to explore is how we can generalize the path decomposition of extreme flows in uncapacitated graphs to argue the complexity of CFG with arc capacities. We know that some special cases of capacitated MCCNFP is polynomial solvable, such as the constant capacitated lot-sizing problem.

3. Solving CFG with fixed-charge costs in practice. When the cost function is the fixed-charge type, CFG can also be formulated as a MIP. Although our DP runs in polynomial time, currently it is more realistic to use MIP solvers to attack CFG instances of large size. It would be interesting to explore how to leverage the theoretical insight gained in this chapter to derive stronger formulations and design more efficient

algorithms for CFG.

4. CFG under uncertainty. In practice, the parameters of CFG, such as the supply and demand at each time period and the cost over each arc, may not be known exactly. It is a challenging task to build a high-fidelity model for CFG when uncertainty is involved.

CHAPTER IV

SELL OR HOLD: A SIMPLE TWO-STAGE STOCHASTIC COMBINATORIAL OPTIMIZATION PROBLEM

4.1 Introduction

An investor owns n indivisible assets and wants to sell k of them by the end of next year. The current market price for each asset is known. Next year's market price for each asset is random, but we assume that the distribution of the price vector is known in advance. The investor needs to decide which assets to sell this year. Then after having observed next year's price vector, he needs to decide which of the remaining assets to sell subject to at most k assets in total are sold. The investor's goal is to maximize the sum of the revenue obtained this year and the expected revenue of next year. We call this problem the sell or hold problem (SHP).

Stochastic programming [36] is widely used to deal with decision problems with uncertain data. When random parameters are introduced, many two-stage stochastic combinatorial optimization problems become NP-hard even if the original deterministic decision problems are easy. For example, the deterministic version of SHP is to decide which k out of n assets to sell to maximize revenue, and the optimal strategy is simply to sell the k most expensive ones. Several two-stage stochastic combinatorial optimization problems have been studied in the literature, such as maximum weighted matching [76], shortest path, vertex cover, bin packing, facility location, set covering [87], steiner trees [69, 70], and spanning trees [52]. Various approximation algorithms based on techniques such as LP rounding, and the primal-dual method have been proposed.

The SHP resembles a market trading problem, but our interest arises from trying to understand the complexity brought about by adding a random element to a very simple combinatorial optimization problem. Indeed we show that a trivial two-stage deterministic problem becomes NP-hard by including a simple random component. Nevertheless we also show that some cases of the stochastic problem are solvable in polynomial time and give approximation results for the general NP-hard problem. The remainder of this chapter is organized as follows. Section 4.2 introduces two equivalent formulations for SHP. Section 4.3 shows that SHP is NP-hard when the second-stage prices are discretely distributed. Section 4.4 shows that SHP is polynomially solvable when the number of scenarios at the second stage is constant. A simple $\max\{1/2, k/n\}$-approximation algorithm and a tight example are presented in Section 4.5. Section 4.6 concludes this chapter. The result in this chapter is a joint work with with Shabbir Ahmed and George Nemhauser and appeared in [72].

4.2 Two Formulations for SHP

4.2.1 A two-stage stochastic programming model

Let n be the total number of assets and k be the number of assets to sell. The current price for asset $i (1 \le i \le n)$ is r_i and the next-year price is $c_i(\omega)$, where $c(\omega) = (c_1(\omega), \ldots, c_n(\omega))$ is an n-dimensional random vector defined on a probability space $(\Omega, \mathcal{F}, P)$. WLOG, assume that r_i and $c_i(\omega)$ are all nonnegative. Let $x_i = 1$ ($x_i = 0$) denote the decision to sell (hold) asset i this year, and $y_i = 1$ ($y_i = 0$) denote the decision to sell (hold) asset i next year. Then SHP can be formulated as a two-stage stochastic integer programming problem:

$$\begin{array}{lll} \max & \sum_{i=1}^n r_i x_i + \mathbb{E}[Q(x, c(\omega))] & \\ \text{s.t.} & \sum_{i=1}^n x_i \le k, & \\ & x_i \in \{0,1\}, & i = 1, \ldots, n, \end{array} \tag{17}$$

where $Q(x, c(\omega))$ is the second-stage value function:

$$\begin{array}{rlll} Q(x, c(\omega)) = & \max & \sum_{i=1}^n c_i(\omega) y_i & \\ & \text{s.t.} & \sum_{i=1}^n y_i = k - \sum_{i=1}^n x_i, & \\ & & y_i \le 1 - x_i, & i = 1, \ldots, n, \\ & & y_i \in \{0,1\}, & i = 1, \ldots, n. \end{array} \tag{18}$$

Observe that in (18), the constraint matrix is totally unimodular (TU). Therefore, whenever x_i is integral for each i, the integrality restriction on y_i is redundant. Due to the constraint

$y_i \leq 1 - x_i$, the constraint $y_i \leq 1$ is also redundant. Therefore,

$$\begin{array}{rll} Q(x, c(\omega)) = & \max & \sum_{i=1}^n c_i(\omega) y_i \\ & \text{s.t.} & \sum_{i=1}^n y_i = k - \sum_{i=1}^n x_i, \\ & & 0 \leq y_i \leq 1 - x_i, \qquad\qquad i = 1, \ldots, n. \end{array} \tag{19}$$

Note that $Q(x, c(\omega))$ is a monotone non-increasing concave function of x for every fixed $c(\omega)$ due to the strong duality theorem of linear programming.

4.2.2 A submodular maximization model

SHP can also be formulated as a non-monotone submodular maximization problem with a cardinality constraint. Let $S \subseteq N = \{1, \cdots, n\}$ denote the set of assets to sell at the first stage, the problem can be formulated as

$$\max \{f(S) : S \subseteq N, |S| \leq k\}, \tag{20}$$

where $f(S) = \sum_{i \in S} r_i + \mathbb{E}[g(S, c(\omega))]$, $g(S, c(\omega))$ is the optimal second-stage revenue when the assets in S are sold at the first stage and the second-stage price vector is $c(\omega)$.

Theorem 9. *$f(S)$ is a submodular function.*

Proof. Let $g_c(S)$ be the value of $g(S, c(\omega))$ when $c(\omega) = c$. We first show that $g_c(S)$ is a submodular function. We will show the following inequality holds for any $S \subseteq N$ and $q, r \in N \setminus S$:

$$g_c(S \cup \{r\}) - g_c(S) \geq g_c(S \cup \{r, q\}) - g_c(S \cup \{q\}). \tag{21}$$

Since no more than k items in total can be sold, $g_c(S)$ is well-defined only when $|S| \leq k$. When $|S| \geq k+1$, we assign the values of $g_c(S)$ in the following way. When $|S| = k+1$, let $g_c(S) = \min_{r,q \in S}\{g_c(S \setminus \{r\}) + g_c(S \setminus \{q\}) - g_c(S \setminus \{r, q\})\}$. Then define the values of $g_c(S)$ sequentially for $|S| = k+2, k+3, \ldots, n$ in the same way. Therefore (21) is satisfied automatically when $|S| \geq k-1$ according to the way $g_c(S)$ is defined. When $|S| \leq k-2$, WLOG assume that $S = \{1, \cdots, l\}$ for some $1 \leq l \leq k-2$. We sort the rest of the $(n-l)$ items according to their prices at the second stage in nonincreasing order, i.e., $c_{l+1} \geq c_{l+2} \geq \ldots \geq c_n$. Consider the following four cases.

1. $q \geq k, r \geq k$. Observe that $g_c(S) = \sum_{i=l+1}^{k} c_i$ and $g_c(S \cup \{r\}) = \sum_{i=l+1}^{k-1} c_i$. Thus, $g_c(S \cup \{r\}) - g_c(S) = -c_k$. Similarly, $g_c(S \cup \{r, q\}) = \sum_{i=l+1}^{k-2} c_i$ and $g_c(S \cup \{q\}) = \sum_{i=l+1}^{k-1} c_i$. Thus, $g_c(S \cup \{r, q\}) - g_c(S \cup \{q\}) = -c_{k-1}$. Since $-c_k \geq -c_{k-1}$, (21) is satisfied.

2. $q \geq k, r \leq k$. Observe that $g_c(S \cup \{r\}) = \sum_{i=l+1, i\neq r}^{k} c_i$. Thus, $g_c(S \cup \{r\}) - g_c(S) = -c_r$. Similarly, $g_c(S \cup \{r, q\}) = \sum_{i=l+1, l\neq r}^{k-1} c_i$. Thus, $g_c(S \cup \{r, q\}) - g_c(S \cup \{q\}) = -c_r$. Therefore, (21) is satisfied.

3. $q \leq k, r \geq k$. Observe that $g_c(S \cup \{r, q\}) = \sum_{i=l+1, i\neq q}^{k-1} c_i$ and $g_c(S \cup \{q\}) = \sum_{i=l+1, i\neq q}^{k} c_i$. Thus, $g_c(S \cup \{r, q\}) - g_c(S \cup \{q\}) = -c_k$. While $g_c(S \cup \{r\}) - g_c(S) = -c_k$, (21) is satisfied.

4. $q \leq k, r \leq k$. Observe that $g_c(S \cup \{r\}) = \sum_{i=l+1, i\neq r}^{k} c_i$. Thus, $g_c(S \cup \{r\}) - g_c(S) = -c_r$. Similarly, $g_c(S \cup \{r, q\}) = \sum_{i=l+1, i\neq r,q}^{k} c_i$ and $g_c(S \cup \{q\}) = \sum_{i=l+1, i\neq q}^{k} c_i$. Thus, $g_c(S \cup \{r, q\}) - g_c(S \cup \{q\}) = -c_r$. Therefore, (21) is satisfied.

Therefore, (21) is satisfied for any $|S| \leq k-2$ and $q, r \in N \setminus S$.

Since integration preserves submodularity, $\mathbb{E}[g(S, c(\omega))] = \int_\Omega g(S, c(\omega)) dP(\omega)$ is submodular. The function $f(S)$ is the sum of a modular function and a submodular function, so it is also submodular. □

Therefore, SHP can be formulated as a submodular maximization problem with a cardinality constraint. Notice that $f(S)$ is neither monotone nor symmetric. An intuitive explanation for non-monotonicity is that selling more assets at the first stage does not guarantee more or less revenue in total.

4.3 Complexity of SHP

Before addressing the complexity of SHP, we need to discuss how the input is represented, i.e., how to encode the uncertain information of the second-stage price $c(\omega)$. In the rest of the chapter, we mainly consider the case where $c(\omega)$ is an n-dimensional discrete random vector with finite support. Assume that $c(\omega)$ could attain m values $\{c_j\}_{j=1}^m$ where $c_j = [c_{1j}, \ldots, c_{nj}]^T$ and $\Pr[c(\omega) = c_j] = p_j$ for $1 \leq j \leq m$. Let $y_{ij} = 1$ ($y_{ij} = 0$) denote the decision to sell (hold) asset i at the second stage when $c(\omega) = c_j$. We call this case the discrete sell or hold problem (DSHP). The expectation in (17) can be expressed as a finite sum and DSHP can be formulated as the integer programming problem

$$\begin{array}{lll} \max & \sum_{i=1}^n r_i x_i + \sum_{j=1}^m p_j \sum_{i=1}^n c_{ij} y_{ij} & \\ \text{s.t.} & \sum_{i=1}^n x_i + \sum_{i=1}^n y_{ij} = k, & j = 1, \ldots, m, \\ & x_i + y_{ij} \leq 1, & i = 1, \ldots, n,\ j = 1, \ldots, m, \\ & x_i \in \{0, 1\}, y_{ij} \geq 0, & i = 1, \ldots, n,\ j = 1, \cdots, m. \end{array} \tag{22}$$

Theorem 10. *DSHP is NP-hard.*

Proof. We show that DSHP is NP-hard by a polynomial reduction from the NP-hard uncapacitated facility location (UFL) problem. UFL is stated as follows. Let $F = \{1, \ldots, n\}$ be a set of facilities and $C = \{1, \ldots, m\}$ be a set of clients. Let f_i be the setup cost of opening facility i and d_{ij} be the transportation cost of assigning client j to facility i. The objective is to find a subset $I \subseteq F$ of facilities to open and a function $\phi : C \to I$ assigning clients to facilities such that the sum of setup cost and transportation cost is minimized.

Given an instance of UFL, we construct an instance of DSHP by letting $\{1, \ldots, n\} = F$ be the set of assets to sell and $\{1, \ldots, m\} = C$ be the set of scenarios at the second stage. Let $k = n - 1$ be the maximum number of assets that can be sold. Set $p_j = 1/m$, $c_{ij} = md_{ij}$ and $r_i = f_i + \sum_{j=1}^m d_{ij}$. Observe that we will sell as many items as we can at the second stage since $c_{ij} \geq 0$, so when $k = n-1$, there will be exactly one item unsold at each scenario. Therefore, we can use a pair (I, ψ) to denote a feasible solution of DSHP, where $I \subseteq F$ and $\psi : C \to F \setminus I$. A solution (I, ψ) means that assets in I are sold at the first stage and all remaining assets but asset $\psi(j)$ are sold in scenario j at the second stage.

Claim: (I, ϕ) is an optimal solution of UFL if and only if $(F \setminus I, \phi)$ is an optimal solution of DSHP.

The objective value of $(F \setminus I, \phi)$ in DSHP is

$$\begin{aligned}
&\sum_{i \in F \setminus I} r_i + \sum_{j=1}^{m} p_j (\sum_{i \in I} c_{ij} - c_{\phi(j)j}) \\
=&\sum_{i \in F \setminus I} (f_i + \sum_{j=1}^{m} d_{ij}) + \sum_{j=1}^{m} p_j [\sum_{i \in I} (md_{ij}) - md_{\phi(j)j}] \\
=&\sum_{i \in F} (f_i + \sum_{j=1}^{m} d_{ij}) - \sum_{i \in I} (f_i + \sum_{j=1}^{m} d_{ij}) + \sum_{j=1}^{m} \sum_{i \in I} d_{ij} - \sum_{j=1}^{m} d_{\phi(j)j} \\
=&\sum_{i \in F} (f_i + \sum_{j=1}^{m} d_{ij}) - [\sum_{i \in I} f_i + \sum_{j=1}^{m} d_{\phi(j)j}].
\end{aligned}$$

The term $\sum_{i \in I} f_i + \sum_{j=1}^{m} d_{\phi(j)j}$ is exactly the objective value of the solution (I, ϕ) for UFL. Therefore, the objective value of the solution (I, ϕ) for UFL is minimum if and only if the objective value of the solution $(F \setminus I, \phi)$ for DSHP is maximum. □

When $c(\omega)$ is a general random variable, the solution of SHP can be approximated by the solutions of a sequence of DSHP by the sample average approximation method [90]. Convergence is guaranteed as the number of samples goes to infinity. In fact, we can always aggregate scenarios such that the total number of scenarios is at most 2^n. Since the optimal second-stage decision for each scenario is to sell some of the most expensive remaining assets, its value only depends on the order of the second-stage prices $\{c_i(\omega)\}$. Therefore, we can aggregate those scenarios with the same price order into one scenario and adjust the associated parameters accordingly. Suppose that l scenarios have the same price order, then the aggregated scenario has probability $\sum_{j=1}^{l} p_j$, and the second-stage price of asset i in the aggregated scenario is $\sum_{j=1}^{l} p_j c_{ij} / (\sum_{j=1}^{l} p_j)$.

4.4 Polynomially solvable cases

In this section, we give some special cases of SHP that can be solved in polynomial time. We have shown that when $k = n$ the optimal decision is to compare the first-stage price and expected second-stage price of each asset and sell it at the higher price. When k is a constant, the number of possible first-stage decisions is $\binom{n}{k} = O(n^k)$, so the problem is polynomially solvable. Now we study the case when the number of scenarios m at the second stage is constant.

4.4.1 DSHP when $m = 2$

First consider $m = 2$, which has a special structure. By (22), the constraint is:

$$\begin{bmatrix} 1 & & & & 1 & & & & & & & \\ & 1 & & & & 1 & & & & & & \\ & & \ddots & & & & \ddots & & & & & \\ & & & 1 & & & & 1 & & & & \\ 1 & \dots & \dots & 1 & 1 & \dots & \dots & 1 & & & & \\ 1 & \dots & \dots & 1 & & & & & 1 & \dots & \dots & 1 \\ 1 & & & & & & & & 1 & & & \\ & 1 & & & & & & & & 1 & & \\ & & \ddots & & & & & & & & \ddots & \\ & & & 1 & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ y_{11} \\ \vdots \\ y_{n1} \\ y_{12} \\ \vdots \\ y_{n2} \end{bmatrix} \leq \begin{bmatrix} 1 \\ \vdots \\ 1 \\ k \\ k \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{23}$$

Let A denote the constraint matrix. We name the constraints in (23) with right hand side k and 1 the k-constraints and 1-constraints, respectively. The fact that we have relaxed the two k-constraints to inequalities does not affect the optimal solutions due to the non-negativity of the prices. Let P_{LP} be the linear programming relaxation of (23). We first present a proposition regarding the constraint matrix of DSHP when $m = 2$.

Proposition 14. *The constraint matrix A is a network matrix.*

Proof. Construct a directed tree (N, A_1) and a digraph (N, A_2) as shown in Figure 20. The vertex set $N = \cup_{i=1}^{n} U_i \cup_{i=1}^{n} V_i \cup_{i=1}^{3} W_i$. We will show that A is an arc-dipath incidence matrix where the corresponding arcs are exactly arcs in A_1 and the paths are given by the endpoints of the arcs in A_2. Let arcs in A_1 correspond to the rows of A. In detail, (U_i, W_1) correspond to the i-th row of A for $i = 1, \dots, n$, (W_1, W_0) and (W_0, W_2) correspond to the $(n+1)$-th and $(n+2)$-th row of A, and (W_2, V_i) correspond to $(n+2+i)$-th row of A for $i = 1, \dots, n$, respectively. Then each column of A is a characteristic vector of certain path whose endpoints are given by arcs in A_2. In detail, (U_i, V_i), (U_i, W_0) and (W_0, V_i) correspond to the i-th, $(n+i)$-th and $(2n+i)$-th columns of A for $i = 1, \dots, n$, respectively. Thus, A is a network matrix. □

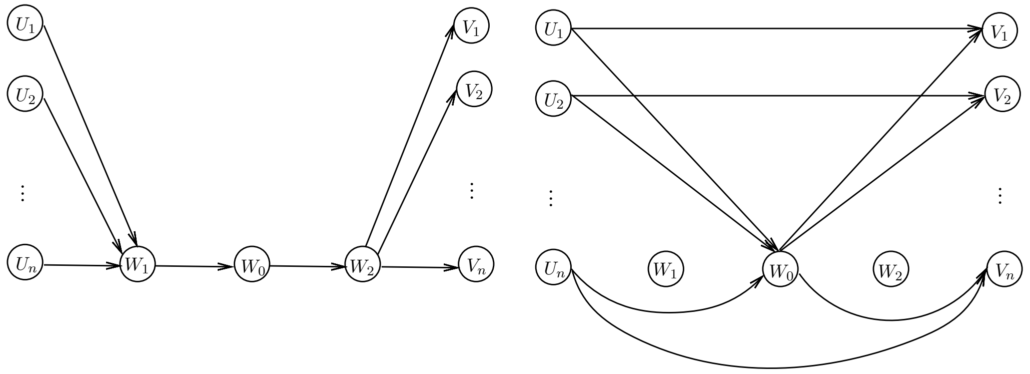


Figure 20: The digraph (N, A_1) and (N, A_2)

Corollary 2. *DSHP with $m = 2$ is solvable in polynomial time.*

Proof. Since A is a network matrix by Proposition 14 and the right-hand side of constraint (23) is integral, the polytope P_{LP} with constraints (23) with all variables in $[0, 1]$ has all of its extreme points integral. □

Remark 2. A linear program $\max\{a^T x | Ax \leq b, x \geq 0\}$ where A is a network matrix can be modeled as a network flow problem [82]. Thus we could solve DSHP when $m = 2$ by a network simplex algorithm. However, we are not aware of any simple combinatorial algorithm that directly solves DSHP when $m = 2$.

When $m \geq 3$ and $n \geq 3$, A is not TU, since it contains the following submatrix

$$\overline{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & & \\ 1 & 1 & 1 & & 1 & \\ 1 & 1 & 1 & & & 1 \\ 1 & & & 1 & & \\ & 1 & & & 1 & \\ & & 1 & & & 1 \end{bmatrix}.$$

The first three columns correspond to decision variables x_1, x_2 and x_3 at the first stage, and the last three columns correspond to decision variables y_{11}, y_{22} and y_{33} at the second stage. The first three rows correspond to the k-constraints for three different scenarios at the second stage, and the last three rows correspond to the 1-constraints for asset 1 at scenario 1, asset 2 at scenario 2 and asset 3 at scenario 3. Since $\det(\overline{A}) = 2$, A is not TU.

4.4.2 DSHP when m is constant

We present a technical lemma first.

Lemma 6. *The integer programming (IP) problem* $(P) : \max\{a^T x | Bx \leq b, x \in \{0,1\}^n\}$, *where the matrix* B *has* m *rows and each entry of* B *is either* 0, 1 *or* -1, *can be solved in* $O(n^{3^m})$ *time.*

Proof. Since each entry of B is either 0 or ± 1, B has at most $M = 3^m$ different columns. Group the variables together if the corresponding columns in B are identical, then there are at most M groups. Suppose that there are n_l variables in the l-th group. Assume that in the objective function, the corresponding coefficients satisfy $a_{l_1} \geq a_{l_2} \geq \ldots \geq a_{l_{n_l}}$. Then any optimal solution of (P) satisfies the condition $x_{l_1} \geq x_{l_2} \geq \ldots \geq x_{l_{n_l}}$. Thus there are at most $n_l + 1$ possible values for these variables of any optimal solution, i.e., $x_{l_1} = \ldots = x_{l_{n_l}} = 0$, and $x_{l_1} = 1, x_{l_2} \ldots = x_{l_{n_l}} = 0$,..., and $x_{l_1} = \ldots = x_{l_{n_l}} = 1$. Therefore, the number of possible values for an optimal solution is bounded by

$$\prod_{l=1}^{M}(n_l + 1) \leq (\frac{\sum_{l=1}^{M}(n_l + 1)}{M})^M = (1 + \frac{n}{M})^M.$$

The last equality follows from the fact that $\sum_{l=1}^{M} n_l = n$. Then the optimal solution can be found by enumeration in $O(n^M) = O(n^{3^m})$ time. □

Theorem 11. *DSHP is polynomially solvable when* m *is constant.*

Proof. The proof is divided into two steps: (a) DSHP can be decomposed into at most k^m IPs, (b) each IP is polynomially solvable.

From (22), once x is fixed, the problem can be decomposed into m optimization problems:

$$\begin{aligned} Q^j(x) = \quad & \max \quad \textstyle\sum_{i=1}^{n} c_{ij} y_{ij} \\ & \text{s.t.} \quad \textstyle\sum_{i=1}^{n} y_{ij} = k - \sum_{i=1}^{n} x_i, \\ & \qquad\;\; 0 \leq y_{ij} \leq 1 - x_i \qquad\qquad i = 1, \ldots, n. \end{aligned} \tag{24}$$

Problem (24) is a *fractional knapsack problem* with unit weight for each item. Suppose σ_j is a permutation of $\{1, \ldots, n\}$ such that $c_{\sigma_j(1)j} \geq c_{\sigma_j(2)j} \geq \ldots \geq c_{\sigma_j(n)j}$. Then the optimal solution of (24) is

$$y_{\sigma_j(i)j} = \begin{cases} 1 - x_{\sigma_j(i)} & i < l_j \\ k + 1 - l_j - \sum_{i=l_j}^{n} x_{\sigma_j(i)} & i = l_j \\ 0 & i > l_j \end{cases} \tag{25}$$

where $l_j = \min\left\{s \middle| \sum_{i=1}^{s} (1 - x_{\sigma_j(i)}) \geq k - \sum_{i=1}^{n} x_i\right\}$. Therefore,

$$Q^j(x) = \sum_{i=1}^{n} c_{\sigma_j(i)j} y_{\sigma_j(i)j} = \sum_{i=1}^{l_j - 1} c_{\sigma_j(i)j}(1 - x_{\sigma_j(i)}) + c_{\sigma_j(l_j)j}(k + 1 - l_j - \sum_{i=l_j}^{n} x_{\sigma_j(i)})$$

The value of l_j depends on the value of x, and the possible values l_j could attain are $\{1, \ldots, k\}$. Hence from (24), $Q^j(x)$ is a concave piecewise linear function of x with at most k pieces, and the objective function of DSHP

$$F(x) = \sum_{i=1}^{n} r_i x_i + \sum_{j=1}^{n} p_j Q^j(x)$$

is a piecewise linear function of x with at most k^m pieces. Therefore, DSHP can be decomposed into at most k^m IPs.

The domain of each piece when $(l_1, \ldots, l_m) = (a_1, \ldots, a_m)$ is determined by

$$\begin{aligned} &\sum_{i=1}^{a_j - 1} (1 - x_{\sigma_j(i)}) < k - \sum_{i=1}^{n} x_j, \ j = 1, \ldots, m, \\ &\sum_{i=1}^{a_j} (1 - x_{\sigma_j(i)}) \geq k - \sum_{i=1}^{n} x_j, \ j = 1, \ldots, m. \end{aligned} \tag{26}$$

Since $F(x)$ is concave, it is continuous in the interior of $[0, 1]^n$. Changing the sign from "<" to "≤" in (26) will not affect the optimal value of DSHP. Then optimizing $F(x)$ over each piece is formulated as

$$\begin{array}{ll} \min & F(x) \\ \text{s.t.} & \sum_{i=1}^{a_j - 1} (1 - x_{\sigma_j(i)}) \leq k - \sum_{i=1}^{n} x_j, \quad j = 1, \ldots, m, \\ & \sum_{i=1}^{a_j} (1 - x_{\sigma_j(i)}) \geq k - \sum_{i=1}^{n} x_j, \quad j = 1, \ldots, m, \\ & \sum_{i=1}^{n} x_i \leq k, \\ & x \in \{0, 1\}^n. \end{array} \tag{27}$$

Problem (27) is an IP with $2m + 1$ constraints, and each entry of the constraint matrix is either 0, 1 or -1. By Lemma 6, each IP can be solved in $O(n^{3^{2m+1}})$ time. Therefore, DSHP can be solved in $O(k^m n^{3^{2m+1}})$ time. When m is constant, DSHP is polynomially solvable. □

4.5 A $\max\{k/n, 1/2\}$*-approximation algorithm for DSHP*

In Section 4.2 we have shown that SHP can be formulated as a submodular maximization problem with a cardinality constraint. Recently, there has been many results on approximation algorithm for submodular maximization [28, 56, 68, 77, 97]. These algorithms could be applied to our problem when they can deal with the additional cardinality constraint. However, these algorithms are designed for general submodular functions given by a value oracle, and therefore their performances are very weak. For example, it has been shown that non-monotone submodular maximization with a matroid independence constraint is hard to approximate to within a factor of $(1/2 + \epsilon)$ for any fixed $\epsilon > 0$, unless P = NP [97].

In contrast, we explore the special structure of SHP and design a simple approximation algorithm that achieves a better approximation ratio. Let $\bar{c}_i = \mathbb{E}[c_i(\omega)]$ denote the expected price of asset i at the second stage.

Algorithm 1 Greedy heuristic 1

```
Set $\bar{r}_i = \max\{r_i, \bar{c}_i\}$ for each asset $i$.
Sort $\bar{r}_1, \dots, \bar{r}_n$ in nonincreasing order.
Let $\{i_1, \dots, i_k\}$ be the top $k$ assets in the list. If $r_{i_l} > \bar{c}_{i_l}$, then sell asset $i_l$ at the first stage, $l = 1, \dots, k$.
For the second stage of each scenario, sort the prices of the remaining assets in nonincreasing order, and sell the top $k - t$ assets, where $t$ is the number of assets sold at the first stage.
```

Proposition 15. *Algorithm 1 is a k/n-approximation algorithm for DSHP.*

Proof. Let $\mathrm{OPT}(I)$ be the optimal objective value for the instance I, and $\mathrm{ALG}_1(I)$ be the objective value of the solution produced by Algorithm 1. Then,

$$\mathrm{ALG}_1(I) \geq \sum_{l=1}^{k} \bar{r}_{i_l} \geq \frac{k}{n} \sum_{i=1}^{n} \bar{r}_i \geq \frac{k}{n} \mathrm{OPT}(I).$$

The last inequality follows from the fact that $\sum_{i=1}^{n} \bar{r}_i$ is the optimal value of DSHP when $k = n$. □

A tight example for Algorithm 1 is obtained by letting the price for each asset at the first stage be $r+\epsilon$. The second stage has n scenarios, each with probability $1/n$. For scenario 1, the price for assets 1 to k is nr/k and the rest of the assets all have price 0. For scenario 2, the price for assets 2 to $k+1$ is nr/k and the rest of the assets all have price 0. For scenario j, the price for assets j to $(j+k) \pmod n$ is nr/k and the rest of the assets have price 0. The expected price for each asset at the second stage is r, which is less than its first-stage price $r+\epsilon$. Thus the solution of Algorithm 1 is to sell k items at the first stage, and $\text{ALG}_1(I) = k(r+\epsilon)$. However, the optimal solution is to sell nothing at the first stage and sell the k most expensive assets at each scenario at the second stage. The optimal objective value $\text{OPT} = nr$.

Algorithm 2 Greedy heuristic 2

```
1: Compute the profit $R_1$ of selling the $k$ most expensive assets in the first stage.
2: Compute the profit $R_2$ of selling the $k$ most expensive assets at each scenario at the
   second stage.
3: If $R_1 \geq R_2$, then sell the $k$ most expensive assets in the first stage, otherwise sell the $k$
   most expensive assets at each scenario at the second stage.
```

Proposition 16. *Algorithm 2 is a* $1/2$*-approximation algorithm for DSHP.*

Proof. Let $\text{ALG}_2(I)$ be the objective value of the solution obtained by Algorithm 2. Then,

$$\text{OPT}(I) = \text{ first-stage profit } + \text{ second-stage profit} \leq R_1 + R_2 \leq 2\text{ALG}_2(I).$$

□

A tight example for Algorithm 2 is obtained by letting $k = 2$. The first-stage price of each asset is 0 except that the price for the first asset is r. The second stage has $n-1$ scenarios, each with probability $1/(n-1)$. For scenario j, the price of each asset is 0 except that the price for asset $j+1$ is r. Then in Algorithm 2, $R_1 = R_2 = r$, and $\text{ALG}_2(I) = \max\{R_1, R_2\} = r$. However, the optimal solution is to sell asset 1 at the first stage, and sell asset $j+1$ at scenario j at the second stage. Then $\text{OPT}(I) = 2r$.

Corollary 3. *If we take the better of the two solutions output by Algorithm 1 and Algorithm 2, the combined algorithm is a* $\max\{k/n, 1/2\}$*-approximation algorithm.*

4.6 Conclusions and future work

In this chapter, we studied the sell or hold problem, a two-stage decision-making problem under uncertainty. We presented two equivalent formulations for the problem: a two-stage stochastic integer programming model, and a non-monotone submodular maximization model with a cardinality constraint. We showed that the problem is NP-hard in general, gave a $\max\{1/2, k/n\}$-approximation algorithm, and identified several polynomially solvable special cases. The complexity result reveals that stochastic combinatorial optimization problems can become very difficult even when the deterministic version is trivial to solve.

From the modeling perspective, one direction of future research is to consider a risk-averse measure such as conditional value at risk as the objective instead of the risk-neutral expectation functional. Another factor to take into account in practice is that usually only historical data of the asset prices are available instead of the full information of the underlying distribution. A distributional robust model would be more appropriate in this situation. From the computational perspective, it would be worthwhile to derive strong valid inequalities for the two-stage SIP formulation of SHP.

REFERENCES

[1] Aggarwal, A. and Park, J. K., "Improved algorithms for economic lot size problems," *Operations Research*, vol. 41, no. 3, pp. 549–571, 1993.

[2] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., *Network flows: theory, algorithms, and applications.* Upper Saddle River: Prentice Hall, 1993.

[3] Andersen, K., Louveaux, Q., Weismantel, R., and Wolsey, L. A., "Inequalities from two rows of a simplex tableau," in *IPCO XII* (Fischetti, M. and Williamson, D. P., eds.), vol. 4513 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, 2007.

[4] Atamtürk, A., "Sequence independent lifting for mixed-integer programming," *Operations Research*, vol. 52, no. 3, pp. 487–490, 2004.

[5] Atamtürk, A. and Küçükyavuz, S., "Lot sizing with inventory bounds and fixed costs: Polyhedral study and computation," *Operations Research*, vol. 53, no. 4, pp. 711–730, 2005.

[6] Atamtürk, A. and Küçükyavuz, S., "An $O(n^2)$ algorithm for lot sizing with inventory bounds and fixed costs," *Operations Research Letters*, vol. 36, no. 3, pp. 297–299, 2008.

[7] Averkov, G., Wagner, C., and Weismantel, R., "Maximal lattice-free polyhedra: finiteness and an explicit description in dimension three," *Mathematics of Operations Research*, vol. 36, no. 4, pp. 721–742, 2011.

[8] Balas, E., "Intersection cuts-a new type of cutting planes for integer programming," *Operations Research*, vol. 19, pp. 19–39, 1971.

[9] Balas, E., Ceria, S., and Cornuéjols, G., "A lift-and-project cutting plane algorithm for mixed 0–1 programs," *Mathematical programming*, vol. 58, no. 1-3, pp. 295–324, 1993.

[10] Balas, E., Ceria, S., and Cornuéjols, G., "Mixed 0-1 programming by lift-and-project in a branch-and-cut framework," *Management Science*, vol. 42, no. 9, pp. 1229–1246, 1996.

[11] Balas, E. and Jeroslow, R. G., "Strengthening cuts for mixed integer programs," *European Journal of Operational Research*, vol. 4, no. 4, pp. 224–234, 1980.

[12] Balas, E. and Margot, F., "Generalized intersection cuts and a new cut generating paradigm," *Mathematical Programming*, vol. 137, no. 1-2, pp. 19–35, 2013.

[13] Basu, A., Bonami, P., Cornuéjols, G., and Margot, F., "Experiments with two-row cuts from degenerate tableaux," *INFORMS Journal on Computing*, vol. 23, no. 4, pp. 578–590, 2011.

[14] Basu, A., Bonami, P., Cornuéjols, G., and Margot, F., "On the relative strength of split, triangle and quadrilateral cuts," *Mathematical Programming*, vol. 126, no. 2, pp. 281–314, 2011.

[15] Basu, A., Campelo, M., Conforti, M., Cornuéjols, G., and Zambelli, G., "On lifting integer variables in minimal inequalities," in *IPCO XIV* (Eisenbrand, F. and Shepherd, B., eds.), vol. 6080 of *Lecture Notes in Computer Science*, pp. 85–95, Springer, June 2010.

[16] Basu, A., Conforti, M., Cornuéjols, G., and Zambelli, G., "Minimal inequalities for an infinite relaxation of integer programs," *SIAM Journal on Discrete Mathematics*, vol. 24, no. 1, pp. 158–168, 2010.

[17] Basu, A., Cornuéjols, G., and Molinaro, M., "A probabilistic analysis of the strength of the split and triangle closures," in *IPCO XV* (Günlük, O. and Woeginger, G. J., eds.), vol. 6655 of *Lecture Notes in Computer Science*, pp. 27–38, Springer, 2011.

[18] Basu, A., Hildebrand, R., and Köppe, M., "Equivariant perturbation in Gomory and Johnson's infinite group problem." preprint, 2012. available at `http://arxiv.org/pdf/1206.2079v1.pdf`.

[19] Basu, A., Hildebrand, R., and Köppe, M., "Equivariant perturbation in Gomory and Johnson's infinite group problem: II. the unimodular two-dimensional case," in *IPCO XVI* (Goemans, M. and Correa, J., eds.), pp. 62–73, Springer, 2013.

[20] Basu, A., Hildebrand, R., Köppe, M., and Molinaro, M., "A $(k+1)$-slope theorem for the k-dimensional infinite group relaxation," *SIAM Journal on Optimization*, vol. 23, no. 2, 2013.

[21] Bellman, R., "Dynamic programming and the smoothing problem," *Management Science*, vol. 3, no. 1, pp. 111–113, 1956.

[22] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A., *Robust Optimization.* Princeton Series in Applied Mathematics, Princeton, New Jersey: Princeton University Press, October 2009.

[23] Bertsekas, D. P., *Dynamic programming and optimal control*, vol. 1. Belmont,Massachusetts: Athena Scientific, 1995.

[24] Bertsekas, D. P. and Tsitsiklis, J. N., *Neuro-dynamic programming.* Belmont, Massachusetts: Athena Scientific, 1996.

[25] Bertsimas, D., Brown, D. B., and Caramanis, C., "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

[26] Birge, J. R. and Louveaux, F., *Introduction to stochastic programming.* Springer Series in Operations Research, New York: Springer Verlag, 1997.

[27] Borozan, V. and Cornuéjols, G., "Minimal valid inequalities for integer constraints," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 538–546, 2009.

[28] Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J., "Maximizing a Submodular Set Function subject to a Matroid Constraint," in *IPCO XII* (Fischetti, M. and Williamson, D. P., eds.), vol. 4513 of *Lecture Notes in Computer Science*, pp. 182–196, Springer, 2007.

[29] Christofides, N., Mingozzi, A., and Toth, P., "State-space relaxation procedures for the computation of bounds to routing problems," *Networks*, vol. 11, no. 2, pp. 145–164, 1981.

[30] Chvátal, V., "Edmonds polytopes and a hierarchy of combinatorial problems," *Discrete Mathematics*, vol. 4, no. 4, pp. 305–337, 1973.

[31] Cook, W., Kannan, R., and Schrijver, A., "Chvátal closures for mixed integer programming problems," *Mathematical Programming*, vol. 47, no. 1-3, pp. 155–174, 1990.

[32] Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., and Schrijver, A., *Combinatorial optimization.* New York: Wiley-Interscience, 1997.

[33] Cornuéjols, G. and Li, Y., "Elementary closures for integer programs," *Operations Research Letters*, vol. 28, no. 1, pp. 1–8, 2001.

[34] Cornuéjols, G. and Margot, F., "On the facets of mixed integer programs with two integer variables and two constraints," *Mathematical Programming*, vol. 120, pp. 429–456, 2009.

[35] Cornuéjols, G. and Molinaro, M., "A 3-slope theorem for the infinite relaxation in the plane," *Mathematical Programming*, pp. 1–23, 2012.

[36] Dantzig, G. B., "Linear programming under uncertainty," *Management Science*, vol. 1, pp. 197–206, 1955.

[37] Dash, S., Dey, S. S., and Günlük, O., "Two dimensional lattice-free cuts and asymmetric disjunctions for mixed-integer polyhedra," *Mathematical programming*, vol. 135, no. 1-2, pp. 221–254, 2012.

[38] Dash, S., Dobbs, N. B., Günlük, O., Nowicki, T. J., and Świrszcz, G. M., "Lattice-free sets, multi-branch split disjunctions, and mixed-integer programming," *Mathematical Programming*, pp. 1–26, 2013.

[39] Dash, S., Günlük, O., and Lodi, A., "MIR closures of polyhedral sets," *Mathematical Programming*, vol. 121, no. 1, pp. 33–60, 2010.

[40] Dash, S., Günlük, O., and Vielma, J. P., "Computational experiments with cross and crooked cross cuts." preprint, 2011. available at `http://www.optimization-online.org/DB_FILE/2011/06/3071.pdf`.

[41] De Farias, D. P. and Van Roy, B., "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

[42] De Farias, D. P. and Van Roy, B., "On constraint sampling in the linear programming approach to approximate dynamic programming," *Mathematics of operations research*, vol. 29, no. 3, pp. 462–478, 2004.

[43] Dekker, R., Fleischmann, M., Inderfurth, K., and Van Wassenhove, L. N., eds., *Reverse logistics: quantitative models for closed-loop supply chains.* Berlin: Springer Verlag, 2004.

[44] Del Pia, A., Wagner, C., and Weismantel, R., "A probabilistic comparison of the strength of split, triangle, and quadrilateral cuts," *Operations Research Letters*, vol. 39, no. 4, pp. 234–240, 2011.

[45] Dey, S. S., "A note on the split rank of intersection cuts," *Mathematical programming*, vol. 130, no. 1, pp. 107–124, 2011.

[46] Dey, S. S., Lodi, A., Tramontani, A., and Wolsey, L. A., "Experiments with two row tableau cuts," in *IPCO XIV* (Eisenbrand, F. and Shepherd, B., eds.), vol. 6080 of *Lecture Notes in Computer Science*, pp. 424–437, Springer, 2010.

[47] Dey, S. S., Lodi, A., Tramontani, A., and Wolsey, L. A., "On the practical strength of two-row tableau cuts," *INFORMS Journal on Computing*, to appear.

[48] Dey, S. S. and Louveaux, Q., "Split rank of triangle and quadrilateral inequalities," *Mathematics of Operations Research*, vol. 36, no. 3, pp. 432–461, 2011.

[49] Dey, S. S. and Richard, J.-P. P., "Facets of two-dimensional infinite group problems," *Mathematics of Operations Research*, vol. 33, no. 1, pp. 140–166, 2008.

[50] Dey, S. S. and Wolsey, L. A., "Lifting integer variables in minimal inequalities corresponding to lattice-free triangles," in *IPCO XIII* (Lodi, A., Panconesi, A., and Rinaldi, G., eds.), vol. 5035 of *Lecture Notes in Computer Science*, pp. 463–475, Springer, 2008.

[51] Dey, S. S. and Wolsey, L. A., "Two row mixed-integer cuts via lifting," *Mathematical Programming*, vol. 124, no. 1-2, pp. 143–174, 2010.

[52] Dhamdhere, K., Ravi, R., and Singh, M., "On two-stage stochastic minimum spanning trees," in *IPCO XI* (Jünger, M. and Kaibel, V., eds.), vol. 3509 of *Lecture Notes in Computer Science*, pp. 321–334, Springer, 2005.

[53] Erickson, R. E., Monma, C. L., and Veinott, Jr., A. F., "Send-and-split method for minimum-concave-cost network flows," *Mathematics of Operations Research*, vol. 12, no. 4, pp. 634–664, 1987.

[54] Espinoza, D., "Computing with multi-row Gomory cuts," in *IPCO XIII* (Lodi, A., Panconesi, A., and Rinaldi, G., eds.), vol. 5035 of *Lecture Notes in Computer Science*, pp. 214–224, Springer, 2008.

[55] Federgruen, A. and Tzur, M., "A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$ time," *Management Science*, vol. 37, no. 8, pp. 909–925, 1991.

[56] Feige, U., Mirrokni, V., and Vondrak, J., "Maximizing non-monotone submodular functions," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 461–471, IEEE, 2007.

[57] Florian, M. and Klein, M., “Deterministic production planning with concave costs and capacity constraints,” *Management Science*, vol. 18, no. 1, pp. 12–20, 1971.

[58] Fukasawa, R. and Günlük, O., “Strengthening lattice-free cuts using non-negativity,” *Discrete Optimization*, vol. 8, no. 2, pp. 229–245, 2011.

[59] Gomory, R. E., “Some polyhedra related to combinatorial problems,” *Linear algebra and its applications*, vol. 2, no. 4, pp. 451–558, 1969.

[60] Gomory, R. E. and Johnson, E. L., “Some continuous functions related to corner polyhedra,” *Mathematical Programming*, vol. 3, no. 1, pp. 23–85, 1972.

[61] Gomory, R. E. and Johnson, E. L., “Some continuous functions related to corner polyhedra, II,” *Mathematical Programming*, vol. 3, no. 1, pp. 359–389, 1972.

[62] Gomory, R. E., “Outline of an algorithm for integer solutions to linear programs,” *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–278, 1958.

[63] Gomory, R. E., “An algorithm for the mixed integer problem,” Tech. Rep. Research Memorandum RM-2597, The Rand Corporation, 1960.

[64] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W., “Lifted flow cover inequalities for mixed 0-1 integer programs,” *Mathematical Programming*, vol. 85, no. 3, pp. 439–467, 1999.

[65] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W., “Sequence independent lifting in mixed integer programming,” *Journal of Combinatorial Optimization*, vol. 4, no. 1, pp. 109–129, 2000.

[66] Guisewite, G. M. and Pardalos, P. M., “Minimum concave-cost network flow problems: applications, complexity, and algorithms,” *Annals of Operations Research*, vol. 25, no. 1, pp. 75–99, 1990.

[67] Guisewite, G. M. and Pardalos, P. M., “A polynomial time solvable concave network flow problem,” *Networks*, vol. 23, no. 2, pp. 143–147, 1993.

[68] Gupta, A., Roth, A., Schoenebeck, G., and Talwar, K., “Constrained non-monotone submodular maximization: offline and secretary algorithms,” in *Proceedings of the 6th international conference on Internet and Network Economics*, pp. 246–257, Springer, 2010.

[69] Gupta, A., Ravi, R., and Sinha, A., “An edge in time saves nine: LP rounding approximation algorithms for stochastic network design,” in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 218–227, IEEE, 2004.

[70] Gupta, A., Ravi, R., and Sinha, A., “LP rounding approximation algorithms for stochastic network design,” *Mathematics of Operations Research*, vol. 32, no. 2, pp. 345–364, 2007.

[71] He, Q., Ahmed, S., and Nemhauser, G. L., “A probabilistic comparison of split and type 1 triangle cuts for two-row mixed-integer programs,” *SIAM Journal on Optimization*, vol. 21, p. 617, 2011.

[72] He, Q., Ahmed, S., and Nemhauser, G. L., "Sell or hold: A simple two-stage stochastic combinatorial optimization problem," *Operations Research Letters*, vol. 40, no. 2, pp. 69–73, 2012.

[73] He, Q., Ahmed, S., and Nemhauser, G. L., "Minimum concave cost flow over a grid network," *Mathematical Programming*, submitted.

[74] Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G., and Wolsey, L., *50 Years of Integer Programming 1958–2008*. Berlin: Springer, 2008.

[75] Kaminsky, P. and Simchi-Levi, D., "Production and distribution lot sizing in a two stage supply chain," *IIE Transactions*, vol. 35, no. 11, pp. 1065–1075, 2003.

[76] Kong, N. and Schaefer, A. J., "A factor 1/2 approximation algorithm for two-stage stochastic matching problems," *European Journal of Operational Research*, vol. 172, pp. 740–746, 2006.

[77] Lee, J., Mirrokni, V., Nagarajan, V., and Sviridenko, M., "Non-monotone submodular maximization under matroid and knapsack constraints," in *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pp. 323–332, ACM, 2009.

[78] Li, Y. and Richard, J.-P. P., "Cook, kannan and schrijvers example revisited," *Discrete Optimization*, vol. 5, no. 4, pp. 724–734, 2008.

[79] Lovász, L., "Geometry of numbers and integer programming," in *Mathematical programming: recent developments and applications* (Iri, M. and Tanabe, K., eds.), pp. 177–201, Kluwer Academic Publishers, 1989.

[80] Martin, R. K., Rardin, R. L., and Campbell, B. A., "Polyhedral characterization of discrete dynamic programming," *Operations research*, vol. 38, no. 1, pp. 127–138, 1990.

[81] Martin, R., "Generating alternative mixed-integer programming models using variable redefinition," *Operations Research*, vol. 35, no. 6, pp. 820–831, 1987.

[82] Nemhauser, G. L. and Wolsey, L. A., *Integer and combinatorial optimization*. New York: Wiley-Interscience, 1988.

[83] Nemhauser, G. L. and Wolsey, L. A., "A recursive procedure to generate all cuts for 0–1 mixed integer programs," *Mathematical Programming*, vol. 46, no. 1-3, pp. 379–390, 1990.

[84] Pochet, Y. and Wolsey, L. A., *Production planning by mixed integer programming*. New York: Springer Verlag, 2006.

[85] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons, Inc., 1994.

[86] Rardin, R. L. and Wolsey, L. A., "Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems," *European Journal of Operational Research*, vol. 71, no. 1, pp. 95–109, 1993.

[87] Ravi, R. and Sinha, A., "Hedging uncertainty: Approximation algorithms for stochastic optimization problems," *Mathematical Programming*, vol. 108, no. 1, pp. 97–114, 2006.

[88] Schrijver, A., *Theory of linear and integer programming.* New York: Wiley-Interscience, 1998.

[89] Schrijver, A., *Combinatorial optimization: polyhedra and efficiency*, vol. 24. Berlin: Springer Verlag, 2003.

[90] Shapiro, A., Dentcheva, D., and Ruszczyński, A., *Lectures on stochastic programming: modeling and theory.* Philadelphia: SIAM, 2009.

[91] Toriello, A., "Optimal toll design: a lower bound framework for the asymmetric traveling salesman problem," *Mathematical Programming*, to appear.

[92] Tuy, H., Ghannadan, S., Migdalas, A., and Värbrand, P., "The minimum concave cost network flow problem with fixed numbers of sources and nonlinear arc costs," *Journal of Global Optimization*, vol. 6, no. 2, pp. 135–151, 1995.

[93] Tuy, H., Ghannadan, S., Migdalas, A., and Värbrand, P., "A strongly polynomial algorithm for a concave production-transportation problem with a fixed number of nonlinear variables," *Mathematical Programming*, vol. 72, no. 3, pp. 229–258, 1996.

[94] van den Heuvel, W. and Wagelmans, A. P., "Four equivalent lot-sizing models," *Operations Research Letters*, vol. 36, no. 4, pp. 465–470, 2008.

[95] van Hoesel, S., Romeijn, H. E., Morales, D. R., and Wagelmans, A. P., "Integrated lot sizing in serial supply chains with production capacities," *Management science*, vol. 51, no. 11, pp. 1706–1719, 2005.

[96] Vazirani, V. V., *Approximation algorithms.* New York: Springer, 2004.

[97] Vondrák, J., "Symmetry and approximability of submodular maximization problems," in *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 651–670, IEEE, 2010.

[98] Wagelmans, A., van Hoesel, S., and Kolen, A., "Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case," *Operations Research*, vol. 40, no. 1, pp. 145–156, 1992.

[99] Wagner, H. M. and Whitin, T. M., "Dynamic version of the economic lot size model," *Management science*, vol. 5, no. 1, pp. 89–96, 1958.

[100] Wolsey, L. A., "Facets and strong valid inequalities for integer programs," *Operations Research*, vol. 24, no. 2, pp. 367–372, 1976.

[101] Wolsey, L. A., "Valid inequalities and superadditivity for 0–1 integer programs," *Mathematics of Operations Research*, vol. 2, no. 1, pp. 66–77, 1977.

[102] Wolsey, L. A., *Integer programming.* Wiley-Interscience Series in Discrete Mathematics and Optimization, New York: Wiley-Interscience, 1998.

[103] Zangwill, W. I., “A backlogging model and a multi-echelon model of a dynamic economic lot size production system-a network approach,” *Management Science*, vol. 15, no. 9, pp. 506–527, 1969.

[104] Zhang, M., Küçükyavuz, S., and Yaman, H., “A polyhedral study of multi-echelon lot sizing with intermediate demands,” *Operations Research*, vol. 60, no. 4, pp. 918–935, 2012.